z/OS Communications Server

IP System Administrator’s Commands

Version 1 Release 9

SC31-8781-07

z/OS Communications Server

IP System Administrator’s Commands

Version 1 Release 9

SC31-8781-07

Note:
Before using this information and the product it supports, be sure to read the general information under "Notices" on page 913.

Eighth Edition (September 2007)

This edition applies to Version 1 Release 9 of z/OS (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You may send your comments to the following address.
International Business Machines Corporation
Attn: z/OS Communications Server Information Development
Department AKCA, Building 501
P.O. Box 12195, 3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195

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

Fax (USA and Canada):
1+919-254-1258

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

Internet e-mail:
comsvrcf@us.ibm.com

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

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

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

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

Contents

Chapter 5. Displaying policy-based networking information 605

Chapter 6. Querying and administrating a Domain Name System (DNS) 661

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 supports both IPv6 and IPv4. Unless explicitly noted, information describes IPv4 networking protocol. IPv6 support is qualified within the text.

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 V1R9.

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 V1R9.

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 V1R9 should already be installed and customized for your network. For information about installing, see the *z/OS 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:

- 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 237 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 247 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
  - The TSO PING and z/OS UNIX **ping/oping** commands
  - The TSO RPCINFO and z/OS UNIX **rpcinfo/orpcinfo** commands
  - The TSO TRACERTE and z/OS UNIX **traceroute/otracert** commands
- Chapter 4, "Managing network security," on page 537 describes how to use the **ipsec** command to obtain or modify IP security information in the network.

- Chapter 5, “Displaying policy-based networking information,” on page 605 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 661 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 763 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 811 describes how to use the SNTP daemon.
- Appendix A, “SNMP capability statement,” on page 815 includes the SNMP agent and subagents capability statement for z/OS Communications Server.
- Appendix B, “Management Information Base (MIB) objects,” on page 839 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 881 shows the 3172 attributes and their corresponding MIB variables.
- Appendix D, “SNMP trap types,” on page 883 lists the generic and enterprise-specific trap types that can be received by SNMP.
- Appendix E, “ICMP/ICMPv6 types and codes,” on page 889 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 891 lists the related protocol specifications for TCP/IP.
- “Information APARs and technotes,” lists information APARs for IP and SNA documents.
- “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 two characters of the publication's file name (also called the book name). The higher the number, the more recent the publication. Also, next to the publication titles in the CD-ROM booklet and the readme files, there is an asterisk (*) that indicates whether a publication is new or changed.

How to contact IBM service

For immediate assistance, visit this Web site:

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

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

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

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

If you would like to provide feedback on this publication, see "Communicating Your Comments to IBM" on page 933.

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=*
//SYSTSPRT DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSTSIN  DD *
 NETSTAT CONN
//
```

The Netstat report output is written to the batch job log.

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 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 of the exit routines described in this document are *installation-wide exit routines*. You will see the installation-wide exit routines also called installation-wide exits, exit routines, and exits throughout this document.

The TPF logon manager, although shipped 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.

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

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 USER command is a keyword. The required variable parameter is *user_id*, and the optional variable parameter is *password*. Replace the variable parameters with your own values.

```
►►──USER──user_id──┬──────────┬────────────────────────────────────────►◄
                   └─password─┘
```

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.

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

You must code required operands and values.

Optional values

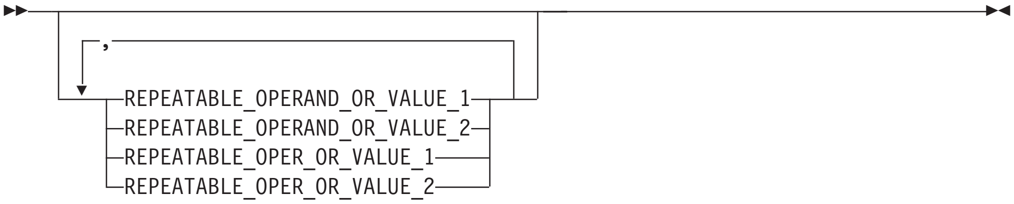
Optional operands and values appear below the main path line.

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

You can choose not to code optional operands and values.

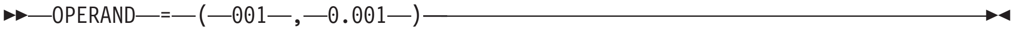
Selecting more than one operand

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

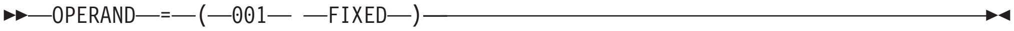
Nonalphanumeric characters

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).



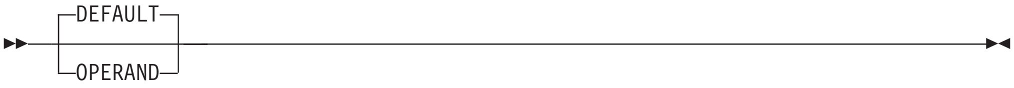
Blank spaces in syntax diagrams

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).



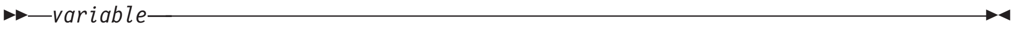
Default operands

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



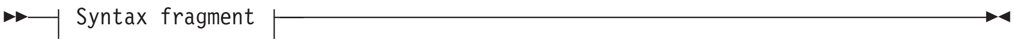
Variables

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.



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:

```
├──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 "z/OS Communications Server information" on page 923, in the back of this document.

Required information

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

Related information

This section contains subsections on:

- "Softcopy information"
- "Other documents" on page xxiii
- "Redbooks" on page xxiii
- "Where to find related information on the Internet" on page xxiv
- "Using LookAt to look up message explanations" on page xxv
- "Using IBM Health Checker for z/OS" on page xxvi

Softcopy information

Softcopy publications are available in the following collections:

Titles	Order Number	Description
z/OS V1R9 Collection	SK3T-4269	This is the CD collection shipped with the z/OS product. It includes the libraries for z/OS V1R9, in both BookManager and PDF formats.
z/OS Software Products Collection	SK3T-4270	This CD includes, in both BookManager and PDF formats, the libraries of z/OS software products that run on z/OS but are not elements and features, as well as the *Getting Started with Parallel Sysplex*® bookshelf.
z/OS V1R9 and Software Products DVD Collection	SK3T-4271	This collection includes the libraries of z/OS (the element and feature libraries) and the libraries for z/OS software products in both BookManager and PDF format. This collection combines SK3T-4269 and SK3T-4270.
z/OS Licensed Product Library	SK3T-4307	This CD includes the licensed documents in both BookManager and PDF format.
IBM System z Redbooks Collection	SK3T-7876	The Redbooks selected for this CD series are taken from the IBM Redbooks inventory of over 800 books. All the Redbooks that are of interest to the zSeries platform professional are identified by their authors and are included in this collection. The zSeries subject areas range from e-business application development and enablement to hardware, networking, Linux, solutions, security, parallel sysplex, and many others.

Other documents

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

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

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

Title	**Number**
DNS and BIND, Fourth Edition, O'Reilly and Associates, 2001	ISBN 0-596-00158-4
Routing in the Internet , Christian Huitema (Prentice Hall PTR, 1995)	ISBN 0-13-132192-7
sendmail, Bryan Costales and Eric Allman, O'Reilly and Associates, 2002	ISBN 1-56592-839-3
SNA Formats	GA27-3136
TCP/IP Illustrated, Volume I: The Protocols, W. Richard Stevens, Addison-Wesley Publishing, 1994	ISBN 0-201-63346-9
TCP/IP Illustrated, Volume II: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63354-X
TCP/IP Illustrated, Volume III, W. Richard Stevens, Addison-Wesley Publishing, 1995	ISBN 0-201-63495-3
TCP/IP Tutorial and Technical Overview	GG24-3376
Understanding LDAP	SG24-4986
z/OS Cryptographic Service System Secure Sockets Layer Programming	*SC24-5901*
z/OS Integrated Security Services LDAP Client Programming	*SC24-5924*
z/OS Integrated Security Services LDAP Server Administration and Use	*SC24-5923*
z/OS JES2 Initialization and Tuning Guide	*SA22-7532*
z/OS Problem Management	*G325-2564*
z/OS MVS Diagnosis: Reference	*GA22-7588*
z/OS MVS Diagnosis: Tools and Service Aids	*GA22-7589*
z/OS MVS Using the Subsystem Interface	*SA22-7642*
z/OS Program Directory	*GI10-0670*
z/OS UNIX System Services Command Reference	*SA22-7802*
z/OS UNIX System Services Planning	*GA22-7800*
z/OS UNIX System Services Programming: Assembler Callable Services Reference	*SA22-7803*
z/OS UNIX System Services User's Guide	*SA22-7801*
z/OS XL C/C++ Run-Time Library Reference	*SA22-7821*
System z9 and zSeries OSA-Express Customer's Guide and Reference	*SA22-7935*

Redbooks

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

Title	Number
Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing	SG24-7339
Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 2: Standard Applications	SG24-7340
Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance	SG24-7341
Communications Server for z/OS V1R8 TCP/IP Implementation, Volume 4: Policy-Based Network Security	SG24-7342
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/servers/eserver/zseries/zos/

z/OS Internet Library

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

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

IBM Communications Server product

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

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

IBM Communications Server product support

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

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

IBM Systems Center publications

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

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

IBM Systems Center flashes

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

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

RFCs

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

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

Internet drafts

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

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

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

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

DNS Web sites

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

USENET news groups
comp.protocols.dns.bind

BIND mailing lists
http://www.isc.org/ml-archives/

BIND Users

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

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

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

Using LookAt to look up message explanations

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

You can use LookAt from these locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux™:

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

- Your z/OS TSO/E host system. You can install code on your z/OS systems to access IBM message explanations using LookAt from a TSO/E command line (for example: TSO/E prompt, ISPF, or z/OS UNIX System Services).
- Your Microsoft® Windows® workstation. You can install LookAt directly from the z/OS Collection (SK3T-4269) or the *z/OS and Software Products DVD Collection* (SK3T-4271) and use it from the resulting Windows graphical user interface (GUI). The command prompt (also known as the DOS > command line) version can still be used from the directory in which you install the Windows version of LookAt.
- Your wireless handheld device. You can use the LookAt Mobile Edition from www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookatm.html with a handheld device that has wireless access and an Internet browser (for example: Internet Explorer for Pocket PCs, Blazer or Eudora for Palm OS, or Opera for Linux handheld devices).

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from:

- A CD-ROM in the z/OS Collection (SK3T-4269).
- The *z/OS and Software Products DVD Collection* (SK3T-4271).
- The LookAt Web site (click **Download** and then select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Using IBM Health Checker for z/OS

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

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

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

How to send your comments

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

- Go to the z/OS contact page at:

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

  There you will find the feedback page where you can enter and submit your comments.
- Send your comments by e-mail to comsvrcf@us.ibm.com. Be sure to include the name of the document, the part number of the document, the version of z/OS

Communications Server, and, if applicable, the specific location of the text you are commenting on (for example, a section number, a page number or a table number).

Summary of changes

Summary of changes
for SC31-8781-07
z/OS Version 1 Release 9

This document contains information previously presented in SC31-8781-06, which supports z/OS Version 1 Release 8.

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.

New information

- Policy-based routing (PBR), see:
  - "Display TCPIP,,NETSTAT" on page 7
  - "DISPLAY TCPIP,,OMPROUTE" on page 18
  - "MODIFY command—OMPROUTE" on page 143
  - "Netstat ALL/-A report" on page 274
  - "Netstat ROUTe/-r report" on page 396
  - Chapter 5, "Displaying policy-based networking information," on page 605
  - Appendix B, "Management Information Base (MIB) objects," on page 839
- Source IP (SRCIP) enhancements, see:
  - "DISPLAY TCPIP,,SYSPLEX" on page 91
  - "Netstat CONFIG/-f report" on page 317
  - "Netstat SRCIP/-J report" on page 417
- Enable AT-TLS for the TN3270E Telnet server, see:
  - "DISPLAY TELNET CONNECTION command" on page 106
  - "VARY QUIESCE command" on page 223
  - "VARY RESUME command" on page 224
  - "VARY STOP command" on page 225
- IPSec network security services, see:
  - "MODIFY command—network security services server" on page 142
  - Chapter 4, "Managing network security," on page 537
- SMTP enhancements, see "MODIFY command—SMTP" on page 161
- MLDv2 and IGMPv3 support, see:
  - "Netstat ALL/-A report" on page 274
  - "Netstat DEvlinks/-d report" on page 345

- Enhance Netstat ALL/-A report to indicate sockets storage use, see "Netstat ALL/-A report" on page 274
- Dynamic LAN idle timer function, see "Netstat DEvlinks/-d report" on page 345
- IPv6 scoped address architecture API, see:
  - "The TSO PING command—Send an echo request" on page 500
  - "The z/OS UNIX ping command—Send an echo request" on page 507
  - "The TSO TRACERTE command—Debug network problems" on page 522
  - "The z/OS UNIX traceroute command—Debug network problems" on page 529
  - "Resolver related commands" on page 661
  - "rndc—Remote control of name server" on page 757
  - "The z/OS UNIX snmp command " on page 764
  - "Using SNMP from NetView" on page 773
  - "The NetView SNMP command" on page 774
- Ping command detection of network MTU, see:
  - "The TSO PING command—Send an echo request" on page 500
  - "The z/OS UNIX ping command—Send an echo request" on page 507

Changed information

- Enable application identifier in NMI, SMF, and Netstat, see:
  - Chapter 1, "Operator commands and system administration," on page 1
  - Chapter 3, "Monitoring the TCP/IP network," on page 247
  - "Netstat ALL/-A report" on page 274
- Allow the TN3270E Telnet server only in a separate address space, see:
  - "DISPLAY TCPIP,,HELP" on page 3
  - "DISPLAY TCPIP,*tnproc*,HELP" on page 94
  - "Display TCPIP,,NETSTAT" on page 7
  - "DISPLAY TCPIP,*tnproc*,TELNET" on page 96
  - "VARY TCPIP,*tnproc*,HELP" on page 218
  - "VARY TCPIP,*tnproc*,TELNET" on page 220
  - "Netstat TELnet/-t report" on page 434
- OMPROUTE enhancements, see:
  - "DISPLAY TCPIP,,OMPROUTE" on page 18
  - "MODIFY command—OMPROUTE" on page 143
- TN3270E Telnet server USSMSG10 client timeout, see:
  - "DISPLAY Telnet CLientID command" on page 97
  - "DISPLAY Telnet PROFILE command" on page 104
  - "DISPLAY TELNET CONNECTION command" on page 106
- Centralized policy services, see:
  - "MODIFY command—Policy Agent" on page 155
  - "The z/OS UNIX pasearch command—Display policies" on page 606
- Support for WLM routing service enhancements for zAAP and zIIP, see:
  - "MODIFY command—z/OS Load Balancing Advisor" on page 178
  - "Netstat ALL/-A report" on page 274
  - "Netstat VDPT/-O report" on page 462
  - "Netstat VIPADCFG/-F report" on page 475

- OSA-Express2 network traffic analyzer enhancements, see:
  - “VARY TCPIP,,OSAENTA” on page 195
  - “Netstat DEvlinks/-d report” on page 345
- VARY TCPIP,,SYSPLEX enhancements, see “VARY TCPIP,,SYSPLEX” on page 210.
- Dynamic VIPA usability enhancements, see:
  - “Netstat CONFIG/-f report” on page 317
  - “Netstat VIPADCFG/-F report” on page 475
- zIIP Exploitation for IPSec, see:
  - “Netstat CONFIG/-f report” on page 317
  - “Netstat STATS/-S report” on page 419
- Add WEIGHTEDACTIVE distribution method for Sysplex Distributor, see:
  - “Netstat VDPT/-O report” on page 462
  - “Netstat VIPADCFG/-F report” on page 475
  - Appendix B, “Management Information Base (MIB) objects,” on page 839

Deleted information

The APPC Application Suite is removed from the z/OS V1R9 Communications Server product and therefore documentation describing APPC Application Suite support has been deleted.

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

You might notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Summary of changes
for SC31-8781-06
z/OS Version 1 Release 8

This document contains information previously presented in SC31-8781-05, which supports z/OS Version 1 Release 7.

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.

New information

- IPv6 support for RPC, see:
  - “MODIFY command—RPCBIND” on page 160
  - “Rpcinfo” on page 515
- Sysplex partitioning, see “DISPLAY TCPIP,,SYSPLEX” on page 91.
- Automated domain name registration, see “MODIFY command—automated domain name registration application (EZBADNR)” on page 114.
- Support for WLM reporting of abnormal conditions, see “MODIFY command—z/OS Load Balancing Advisor” on page 178.
- Netstat enhancements, see:
  - “Display TCPIP,,NETSTAT” on page 7
  - “Netstat” on page 247
- OSA-Express network traffic analyzer support, see “VARY TCPIP,,OSAENTA” on page 195.
- Unreachable DVIPA detection and recovery, see:
  - “Netstat CONFIG/-f report” on page 317
  - “Netstat DEvlinks/-d report” on page 345
  - “Netstat SLAP/-j report” on page 407
  - “Netstat VIPADyn/-v report” on page 489
  - Appendix B, “Management Information Base (MIB) objects,” on page 839
- IPv6 support for integrated IPSec/VPN, see:
  - “Netstat CONFIG/-f report” on page 317
  - “Netstat DEvlinks/-d report” on page 345
  - “IP filter (-f) primary option” on page 560
  - “Manual tunnel (-m) primary option” on page 570
  - “IKE tunnel (-k) primary option” on page 573
  - “Dynamic tunnel (-y) primary option” on page 577
  - “Interface (-i) primary option” on page 591
  - “IP traffic test (-t) primary option” on page 592
  - “The z/OS UNIX pasearch command—Display policies” on page 606
  - Appendix B, “Management Information Base (MIB) objects,” on page 839
- Source IP address selection based on destination address, see “Netstat SRCIP/-J report” on page 417.
- OSA-Express virtual MAC support, see “Netstat DEvlinks/-d report” on page 345.
- Optimized sysplex distributor load balancing and data path, see:
  - “Netstat VDPT/-O report” on page 462
  - “Netstat VIPADCFG/-F report” on page 475
  - Appendix B, “Management Information Base (MIB) objects,” on page 839
- Network address port translation traversal support for integrated IPSec/VPN, see:
  - “IP filter (-f) primary option” on page 560
  - “IKE tunnel (-k) primary option” on page 573
  - “Dynamic tunnel (-y) primary option” on page 577
  - “NATT port translation (-o) primary option” on page 597
- AES cryptographic support for integrated IPSec/VPN, see:
  - “IKE tunnel (-k) primary option” on page 573

  - “Dynamic tunnel (-y) primary option” on page 577
- Support intrusion detection services policy in flat file format, see “The z/OS UNIX pasearch command—Display policies” on page 606.

Changed information

- SNMP enhancements, see:
  - “Display TCPIP,,NETSTAT” on page 7
  - “DISPLAY TCPIP,,Netstat,PORTList report” on page 17
  - “Netstat PORTList/-o report” on page 394
  - “The z/OS UNIX snmp command ” on page 764
  - Appendix B, “Management Information Base (MIB) objects,” on page 839
- TN3270 enhancements, see:
  - “DISPLAY Telnet CLientID command” on page 97
  - “DISPLAY Telnet PROFILE command” on page 104
  - “DISPLAY TELNET CONNECTION command” on page 106
- ARP and ND takeover message enhancements, see “Netstat DEvlinks/-d report” on page 345.

Deleted information

- Support for version 1 networking service level agreement MIB is removed from the V1R8 z/OS Communications Server product, and therefore documentation describing this support has been deleted.
- Support for z/OS Firewall Technologies is removed from the V1R8 z/OS Communications Server product, see:
  - “Netstat CONFIG/-f report” on page 317
  - “The z/OS UNIX trmdstat command—Display traffic regulation management daemon (TRMD) log” on page 626
  - Appendix B, “Management Information Base (MIB) objects,” on page 839
  - “Dynamic tunnel (-y) primary option” on page 577
- Remove ASORTEDPARMS and KEEPALIVEOPTIONS statements from TCP/IP profile configuration, see:
  - “Netstat CONFIG/-f report” on page 317
  - “The z/OS UNIX trmdstat command—Display traffic regulation management daemon (TRMD) log” on page 626

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

You might notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Summary of changes for SC31-8781-05 z/OS Version 1 Release 7

This document contains information previously presented in SC31-8781-04, which supports z/OS Version 1 Release 6.

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

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

New information

- Integrated IPSec/VPN support
  - Policy Agent Support for IPSec/VPN in “The z/OS UNIX pasearch command—Display policies” on page 606.
  - “MODIFY command—IKE server” on page 138.
  - Chapter 4, “Managing network security,” on page 537 describing IPv4 Integrated IPSec/VPN support.
  - IPv4 IP security information, “Netstat CONFIG/-f report” on page 317, “Netstat DEvlinks/-d report” on page 345, and Appendix B, “Management Information Base (MIB) objects,” on page 839.
- Application Transparent Transport Layer Security (AT-TLS)
  - Expanded Netstat support for AT-TLS function. See “Netstat TTLS/-x report” on page 441 and “Display TCPIP,,NETSTAT” on page 7 for displaying AT-TLS information for TCP protocol connections.
  - Policy Agent Support for Application Transparent TLS, see “The z/OS UNIX pasearch command—Display policies” on page 606.
- z/OS Load Balancing Advisor
  - “MODIFY command—z/OS Load Balancing Advisor” on page 178.
  - “MODIFY command—z/OS Load Balancing Agent” on page 186.
- TCP/IP Sysplex operational enhancements
  - JOINgroup, DEACTivate, and REACTivate keywords, see “VARY TCPIP,,SYSPLEX” on page 210, the “Netstat CONFIG/-f report” on page 317, and the “Netstat VIPADCFG/-F report” on page 475.
  - QUIesce and RESUME keywords, see “VARY TCPIP,,SYSPLEX” on page 210, “Netstat ALL/-A report” on page 274, and “Netstat VDPT/-O report” on page 462.

Changed information

- Optimized routing for sysplex distributor, see “Netstat ALL/-A report” on page 274, “Netstat PORTList/-o report” on page 394, “Netstat VDPT/-O report” on page 462, “Netstat VIPADCFG/-F report” on page 475, and Appendix B, “Management Information Base (MIB) objects,” on page 839.
- Server-specific WLM for sysplex distributor
  - Display DISTMethod SERVERWLM parameter, see “Netstat VIPADCFG/-F report” on page 475 and “Netstat VDPT/-O report” on page 462.
  - Display Shareport flag, see “Netstat PORTList/-o report” on page 394.
  - Display WLM weights and shareport distribution type, see “Netstat ALL/-A report” on page 274.

- Sysplex autonomics health monitor for target stacks, see “Netstat ALL/-A report” on page 274, “Netstat VDPT/-O report” on page 462, and Appendix B, “Management Information Base (MIB) objects,” on page 839.
- QDIO OSA-Express2 segmentation offload
  - See “Netstat DEvlinks/-d report” on page 345.
- SNMP IPv6 UDP MIBs support, see “Netstat ALL/-A report” on page 274, “Netstat ALLConn/-a report” on page 298, “Netstat BYTEinfo/-b report” on page 307, “Netstat COnn/-c report” on page 339, “Netstat SOCKets/-s report” on page 411 and Appendix B, “Management Information Base (MIB) objects,” on page 839.
- IPv6 support for HiperSockets
  - See “Netstat ARp/-R report” on page 305, “Netstat CONFIG/-f report” on page 317, “Netstat DEvlinks/-d report” on page 345, and “Netstat ND/-n report” on page 391.
- IPv6 advanced socket APIs
  - See “Netstat ALL/-A report” on page 274.
- Promotion of the use of IPv6 global unicast addresses

  Site-local addresses were designed to use private address prefixes that could be used within a site without the need for a global prefix. Until recently, the full negative impacts of site-local addresses in the Internet were not fully understood. The IETF has deprecated the special treatment given to the site-local prefix. Because of this, it is preferable to use global unicast addresses. This means we are replacing addresses and prefixes that use the site-local prefix (fec0::/10) with ones that use the global prefix for documentation (2001:0DB8::/32). Some samples and examples in this document may display site-local prefixes instead of the now preferred global unicast addresses.

Deleted information

- MODIFY command—OROUTED.

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

You might notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in our documents.

Chapter 1. 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 112
- “VARY TCPIP command” on page 190

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──┬───────────────────────────────┬────────────────────►◄
    └─S─────┘            └─,PARMS='(CTRACE(xxxxxxxx))'───┘
```

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

For more information, 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 either version (BIND.4.9.3, or BIND.9) of 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─┘
```

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.

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, STOR, SYSPLEX	See "DISPLAY command — TCP/IP address space"
TN3270E Telnet server address space	HELP, STOR, TELNET	See "DISPLAY command — TN3270E Telnet server address space" on page 94

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.

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 3
NETSTAT	"Display TCPIP,,NETSTAT" on page 7
OMPROUTE	"DISPLAY TCPIP,,OMPROUTE" on page 18
STOR	"DISPLAY TCPIP,,STOR" on page 90
SYSPLEX	"DISPLAY TCPIP,,SYSPLEX" on page 91

DISPLAY TCPIP,,HELP

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

Format:

```
                                          ┌─,HElp───────┐
►►──Display ──TCPIP──,──┬─────────┬─,HElp─┼─────────────┼──────────────────────────►◄
                        └─tcpproc─┘       ├─,Display────┤
                                          ├─,Vary───────┤
                                          ├─,Obeyfile───┤
                                          ├─,OSAENTA────┤
                                          ├─,DATtrace───┤
                                          ├─,PKTtrace───┤
                                          ├─,PURGECache─┤
                                          ├─,STArt──────┤
                                          ├─,STOp───────┤
                                          ├─,STOR───────┤
                                          ├─,Netstat────┤
                                          ├─,ACCess─────┤
                                          ├─,ALLConn────┤
                                          ├─,ARp────────┤
                                          ├─,BYTEinfo───┤
                                          ├─,CACHinfo───┤
                                          ├─,CONFIG─────┤
                                          ├─,COnn───────┤
                                          ├─,DEvlinks───┤
                                          ├─,DRop───────┤
                                          ├─,HOme───────┤
                                          ├─,IDS────────┤
                                          ├─,ND─────────┤
                                          ├─,PORTList───┤
                                          ├─,ROUTe──────┤
                                          ├─,SOCKets────┤
                                          ├─,SRCIP──────┤
                                          ├─,STATS──────┤
                                          ├─,TTLS───────┤
                                          ├─,VCRT───────┤
                                          ├─,VDPT───────┤
                                          ├─,VIPADCFG───┤
                                          ├─,VIPADyn────┤
                                          ├─,OMProute───┤
                                          ├─,OSPF───────┤
                                          ├─,RIP────────┤
                                          ├─,GENERIC────┤
                                          ├─,RTTABLE────┤
                                          ├─,IPV6OSPF───┤
                                          ├─,IPV6RIP────┤
                                          ├─,GENERIC6───┤
                                          ├─,RT6TABLE───┤
                                          ├─,SYSplex────┤
                                          ├─,LEAVEGROUP─┤
                                          ├─,JOINGROUP──┤
                                          ├─,DEACTIVATE─┤
                                          ├─,REACTIVATE─┤
                                          ├─,QUIesce────┤
                                          └─,RESUME─────┘
```

Parameters:

HElp
Show help on the Display HElp command.

Display
Show help on the Display TCPIP command.

Vary
Show help on the Vary TCPIP command.

Obeyfile
Show help on the Vary Obeyfile command.

DATtrace
Show help on the Vary DATTRACE command.

OSAENTA
Show help on the Vary OSAENTA command.

PKTtrace
Show help on the Vary PKTTRACE command.

PURGECache
Show help on the Vary PURGECache command.

STArt
Show help on the Vary START command.

STOp
Show help on the Vary STOP command.

STOR
Show help on the Display STOR command.

Netstat
Show help on the Display NETSTAT command.

ACCess
Show help on the Display NETSTAT, ACCess, NETWORK command.

ALLConn
Show help on the Display NETSTAT,ALLConn command.

ARp
Show help on the Display NETSTAT,ARP command.

BYTEinfo
Show help on the Display NETSTAT,BYTEinfo command.

CACHinfo
Show help on the Display NETSTAT,CACHinfo command.

CONFIG
Show help on the Display NETSTAT,CONFIG command.

COnn
Show help on the Display NETSTAT,COnn commands.

DEvlinks
Show help on the Display NETSTAT,DEvlinks command.

DRop
Show help on the Vary DRop command.

HOme
Show help on the Display NETSTAT,HOme command.

ND
Show help on the Display NETSTAT,ND command.

IDS
Show help on the Display NETSTAT,IDS command.

PORTList
Show help on the Display NETSTAT,PORTList command.

ROUTe
Show help on the Display NETSTAT,ROUTe command.

SOCKets
Show help on the Display NETSTAT,SOCKets command.

SRCIP
Show help on the Display NETSTAT,SRCIP command.

STATS
Show help on the Display NETSTAT,STATS command.

VCRT
Show help on the Display NETSTAT,VCRT command.

TTLS
Show help on the Display NETSTAT,TTLS command.

VDPT
Show help on the Display NETSTAT,VDPT command.

VIPADCFG
Show help on the Display NETSTAT,VIPADCFG command.

VIPADyn
Show help on the Display NETSTAT,VIPADyn and Display SYSPLEX,VIPADyn commands.

OMProute
Show help on the Display OMPROUTE command.

OSPF
Show help on the Display OMPROUTE,OSPF command.

RIP
Show help on the Display OMPROUTE,RIP command.

GENERIC
Show help on the Display OMPROUTE,GENERIC command.

RTTABLE
Show help on the Display OMPROUTE,RTTABLE command.

IPV6OSPF
Show help on the Display OMPROUTE,IPV6OSPF command.

IPV6RIP
Show help on the Display OMPROUTE,IPV6RIP command.

GENERIC6
Show help on the Display OMPROUTE,GENERIC6 command.

RT6TABLE
Show help on the Display OMPROUTE,RT6TABLE command.

SYSplex
Show help on the Display SYSPLEX and VARY SYSPLEX commands.

LEAVEGROUP
Show help on the Vary SYSPLEX,LEAVEGROUP command.

JOINgroup
Show help on the Vary SYSPLEX,JOINGROUP command.

DEACTIVATE
Show help on the Vary SYSPLEX,DEACTIVATE command.

REACTIVATE
Show help on the Vary SYSPLEX,REACTIVATE command.

QUIesce
Show help on the Vary SYSPLEX,QUIESCE commands.

RESUME
Show help on the Vary SYSPLEX,RESUME commands.

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

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

```
 EZZ0372I D...NETSTAT(,ACCESS|ALLCONN|ARP|BYTEINFO|CACHINFO|
 EZZ0372I CONFIG|CONN|DEVLINKS|HOME|IDS|ND|PORTLIST|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:

```
d tcpip,,help,conn

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

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

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

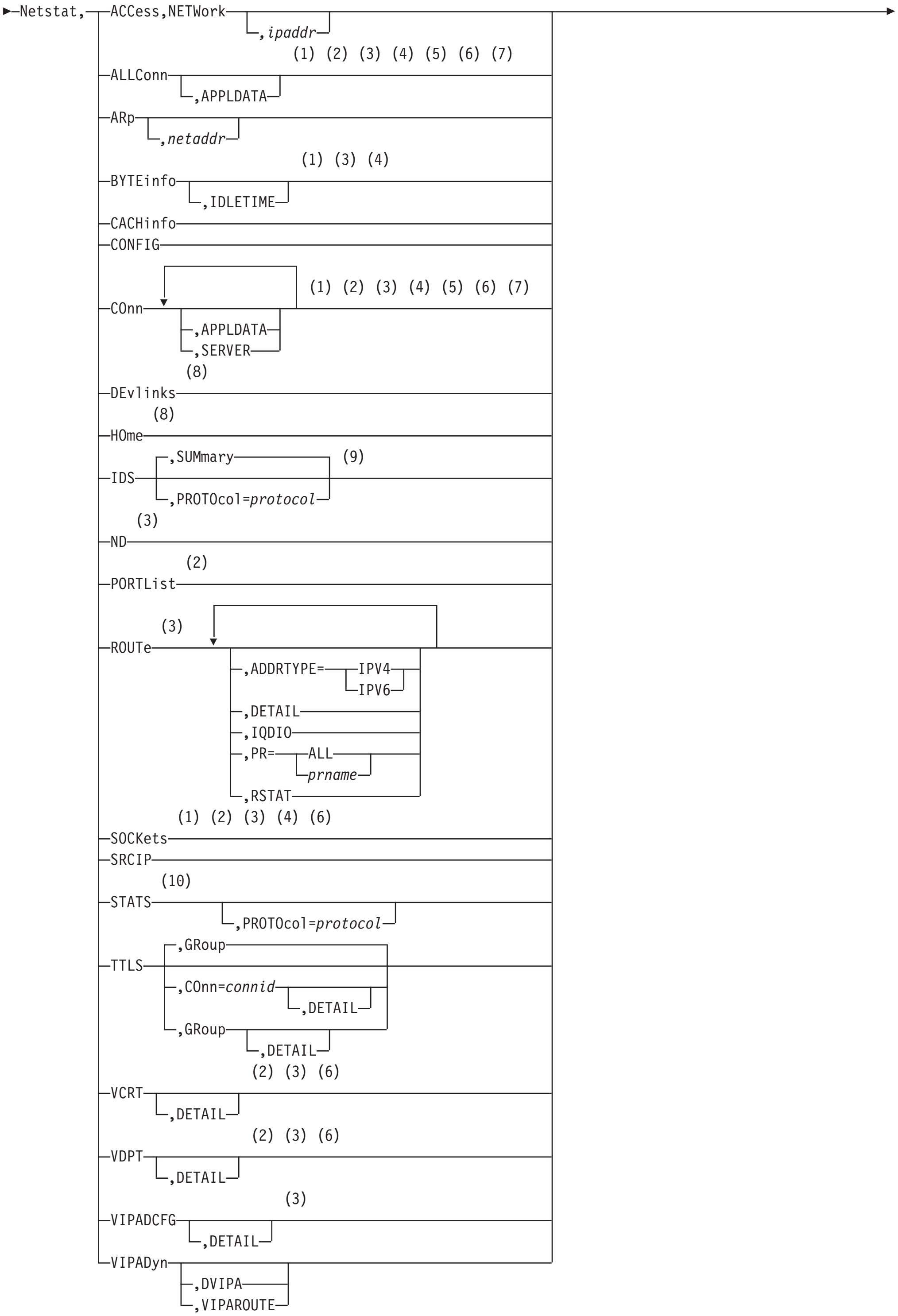
where XDEVNAME is the device name.

Display TCPIP,,NETSTAT

Purpose: 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 269.

Format:

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



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

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

```
                             ┌─,MAX=100──┐ (11)
├──┬──────────────────────┬──┼───────────┼─────────────────────────────────────┤
   └─,FORMat=──┬─LONG──┬──┘  ├─,MAX=*────┤
               └─SHORT─┘     └─,MAX=recs─┘
```

Notes:

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

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

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

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

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

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

7 The APPLD filter is valid only with ALLConn and COnn.

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

9 The valid protocol values are TCP and UDP.

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

11 The MAX filter limits the number of records displayed to the MVS operator's console.

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.

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 idletime 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 will be 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.

DEvlinks
Displays information about devices, links, and interfaces in the TCP/IP address space.

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 port reservation list. For ports reserved by the PORTRANGE profile statement, only one output line is displayed for each range.

Note: The F flag indicates that the displayed port is defined as a SAF resource. For a complete list of other flags, see “Netstat PORTList/-o report” on page 394.

ROUTe

Displays routing information. For a complete description of ROUTE, see “Netstat ROUTe/-r report” on page 396.

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.

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

IQDIO

Displays the HiperSockets™ Accelerator routing table. This parameter is mutually exclusive with the RSTAT parameter.

PR

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

ALL

Displays all policy-based routing tables.

prname

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

Restrictions:

- The PR modifier does not support IPv6 routes. If the PR modifier is used with the ADDRTYPE=IPV6 keyword, no information is displayed.
- Only active policy-based route tables can be displayed with the Netstat ROUTe command. A policy-based route table is active if it is referenced by an active routing rule and its associated action. You can display both 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 606.

RSTAT

Displays all of the 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 IQDIO parameter.

SOCKets

Displays information about each client using the socket interface.

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.

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 298.

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, and routing 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, the output will contain policy action information, target responsiveness values, and a WQ value (on a separate line). If DETAIL is not specified, the output will not contain policy action information or target responsiveness and WQ values.

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 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 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.

POrt=*portnum*
Specifies a port that is used to limit the ALLConn, COnn, PORTList, SOCKets, VCRT, and VDPT options. The port value range is 0–65 535. No wildcards are allowed.

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

ipaddr Provides the response of ALLConn, BYTEinfo, COnn, ND, ROUTe, SOCKets, VCRT, and VDPT on the specified IP address (*ipaddr*). For IPv4 addresses, the default subnet mask `255.255.255.255` is used. For IPv6 addresses, the default prefix length of 128 is used.

ipaddr/subnetmask
Provides the response of 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 of 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 ALLConn, COnn, SOCKets, VCRT, and VDPT report options to the TCP local or 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 ALLConn, BYTEinfo, COnn, and SOCKets, excluding TN3270E Telnet server connections.

INTFName=*intfname*

For DEvlinks and HOme, select information on the specified link or interface name. 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 specified on the LINK profile statement. Otherwise, use the interface name specified on the INTERFACE profile statement.

The INTFName filter can also be used to format a specific OSAENTA trace interface by specifying EZANTA*portname*, where the *portname* value is the name specified on the PORTNAME keyword in the TRLE statement for the OSA that is being traced.

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 specified or in effect by default on the TCPCONFIG statement) and all connections that are not using TCP protocol. For TCP connections that were established while the AT-TLS function was enabled, this includes the following:

- 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. The possible values for AT-TLS connection type include the following:

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/-x 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.

MAX=*recs*

The number of records to be written to the console. The valid range is 1–65 535. This option applies to the ACCess, ALLConn, ARp, BYTEinfo, CACHinfo, COnn, DEvlinks, HOme, IDS, ND, PORTList, ROUTe, SOCKets, SRCIP, VCRT, VDPT, and VIPADyn reports. For the DEvlinks report, the values in the `n OF m RECORDS DISPLAYED` output line do not apply to the LAN group or to the OSA-Express network traffic analyzer information.

The number of records written and the total number of records that could have been written are displayed at the end of the report in the following output line:

```
n OF m RECORDS DISPLAYED
```

Where *n* is the number of records that were written and *m* is the total number of records that could be written.

Examples:

DISPLAY TCPIP,,NETSTAT,ACCESS,NETWORK report:

Purpose: Use the DISPLAY TCPIP,,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
 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
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).

NETWORK PREFIX

Can be one of the following:

- 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 will be 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).

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:

- 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,,Netstat,PORTList report:

Examples: **Not IPv6 enabled (SHORT format)**

```
d tcpip,,netstat,portlist
```

```
EZZ2500I NETSTAT CS V1R9 TCPCS 349
PORT# PROT USER     FLAGS    RANGE       IP ADDRESS      SAF NAME
00020 TCP  OMVS     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
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
17 OF 17 RECORDS DISPLAYED
END OF THE REPORT
```

IPv6 enabled or request for LONG format

d tcpip,,netstat,portlist

```
EZZ2500I NETSTAT CS V1R9 TCPCS 349

PORT# PROT USER     FLAGS    RANGE       SAF NAME
----- ---- ----     -----    -----       --------
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
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
14 OF 14 RECORDS DISPLAYED
END OF THE REPORT
```

Report field descriptions: Descriptions of the DISPLAY TCPIP,,Netstat,PORTList command FLAGS value and the SAF NAME field follow. For descriptions of the other report fields, see "Netstat PORTList/-o report" on page 394.

FLAGS

F Indicates that the displayed port is defined as a SAF resource.

For a complete list of other flag values, see "Netstat PORTList/-o report" on page 394.

SAF NAME

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

DISPLAY TCPIP,,OMPROUTE

Purpose: 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──┬──────────────────────────────────┬─────────┘
                ├─,DEST=──┬─ip_addr───────────┬────┤
                │         └─ip_addr/prefixlen─┘    │
                └─,DELETED─────────────────────────┘
```

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─┘
```

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.

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 route 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 route tables appear in the display of OMPROUTE route tables. A policy-based route 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 contents of an OMPROUTE routing table might contain information that is different from that in a TCP/IP routing table. For more information about displaying the contents of the TCP/IP routing tables, see "Display TCPIP,,NETSTAT" on page 7.

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:

- 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. Following are the 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 uouter.

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 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:

- 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 the OMPROUTE IPv6 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.

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.

- The RT6TABLE parameter displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The contents of the OMPROUTE routing table might contain information that is different from that 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 7.

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 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

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.

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
   ```

Following is 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:

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:

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).

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:

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 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 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 will be 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
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:

1	Down
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

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:

1	Down
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:

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).

# 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).

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:

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
```

NEIGHBOR IP ADDRESS
Displays the neighbor interface IP address.

OSPF ROUTER ID
Neighbor OSPF router ID.

NEIGHBOR STATE
Can be one of the following:

- 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 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 should 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 has been received from the neighbor.

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.

OSPF router routes: The `DISPLAY TCPIP,`*`tcpipjobname`*`,OMPROUTE,OSPF,ROUTERS` command displays all routes to 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 is a sample output with an explanation of entries:

```
EZZ7856I OSPF STATISTICS 380
                OSPF ROUTER ID:         9.167.100.13
                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.

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 will be imported.

IMPORT EXTERNAL ROUTES
Displays the external routes that will be 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 will be included in RIP responses sent out on this interface.

RECEIVE
Specifies which types of routes will be 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 will be 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 contents of the OMPROUTE routing table might contain information that is different from that 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 7.

```
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.

AGE Indicates the time that has elapsed since the routing table entry was last refreshed.

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 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 contents of the OMPROUTE routing table might contain information that is different from that 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 7.

```
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.

Tips:

1. If the route is an OSPF inter-area or intra-area route, this is the OSPF cost of the route.
2. If the route is an OSPF External type 1, this is the OSPF cost to the AS Boundary Router or Forwarding address that is used to reach the destination, plus the external cost.
3. If the route is an OSPF External type 2, this is the external cost.
4. If the route is RIP, this is the RIP metric.
5. If the route is Direct or Static, this cost is irrelevant.

AGE Indicates the time that has elapsed since the routing table entry was last refreshed.

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 contents of the OMPROUTE routing tables might contain information that is different from that 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 7.
- If a policy-based route table is configured with no dynamic routing parameters, OMPROUTE has no knowledge of that route table. The route table does not appear in the display of OMPROUTE policy-based route 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, 0 NETS INACTIVE
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, 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
```

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

INTERFACE
Indicates the name of an interface specified in a dynamic routing parameter for the policy-based routing table. If the interface is not currently defined to the TCP/IP stack or 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 51 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 contents of the OMPROUTE routing table might contain information that is different from that 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 7.

- If a policy-based route table is configured with no dynamic routing parameters, OMPROUTE has no knowledge of that route table. 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 54 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 contents of the OMPROUTE routing table might contain information that is different from that 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 7.
- If a policy-based route table is configured with no dynamic routing parameters, OMPROUTE has no knowledge of that route table. 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 OMPROUTE IPv4 main routing table” on page 53 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 contents of the OMPROUTE routing tables might contain information that is different from that 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 7.
- If a policy-based route table is configured 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.

```
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 OMPROUTE IPv4 main routing table” on page 53 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 51).

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
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

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.

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 will be imported into the IPv6 OSPF domain.

IMPORT EXTERNAL ROUTES
Indicates the types of external routes that will be imported into the IPv6 OSPF domain. Displayed only when AS Boundary Capability is enabled.

ORIG DEFAULT ROUTE
Indicates whether a default route will be 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.

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 will be 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
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:

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:

1	Down
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
```

```
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 interface's 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_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:

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:

1	Down
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 will never become 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 will be 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).

# 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).

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 will be 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:

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:

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 will be 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=*).

Following are 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:

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
```

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:

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 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:

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.

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 should 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 has been received from the neighbor.

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.

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 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 is 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.

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.
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:

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:

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:

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 is 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 69.

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 is 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 69.

PREFIX
The prefix being described by the LSA.

PREFIX OPTIONS
The optional capabilities of the prefix including the following:

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 is 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 69.

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 is 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 69.

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:

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.

Following is 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 69.

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:

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 is 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 69.

# 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:

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.

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

OMPROUTE IPv6 routing table: The DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE command displays all of the routes in the OMPROUTE IPv6 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 contents of the OMPROUTE routing table might contain information that is different from that in the TCP/IP routing table.

```
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.

Tips:

1. If the route is an OSPF inter-area or intra-area route, this is the OSPF cost of the route.

2. If the route is an OSPF External type 1, this is the OSPF cost to the AS Boundary Router or Forwarding address that is used to reach the destination, plus the external cost.
3. If the route is an OSPF External type 2, this is the external cost.
4. If the route is RIP, this is the RIP metric.
5. If the route is Direct or Static, this cost is irrelevant.

AGE
Indicates the time that has elapsed since the routing table entry was last refreshed.

NETS DELETED
Indicates the number of routes that have been deleted from the OMPROUTE routing table and not replaced. Use the D TCPIP,,OMPROUTE,RT6TABLE,DELETED command to list these routes.

NETS INACTIVE
Used for internal debugging purposes only.

IPv6 Route expansion information: Use the DISPLAY TCPIP,*tcpipjobname*,OMPROUTE,RT6TABLE,DEST=*ip_addr* command to obtain information about a particular IPv6 route. 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.

Tips:

1. If the route is an OSPF inter-area or intra-area route, this is the OSPF cost of the route.
2. If the route is an OSPF External type 1, this is the OSPF cost to the AS Boundary Router or Forwarding address that is used to reach the destination, plus the external cost.
3. If the route is an OSPF External type 2, this is the external cost.
4. If the route is RIP, this is the RIP metric.
5. If the route is Direct or Static, this cost is irrelevant.

AGE
Indicates the time that has elapsed since the routing table entry was last refreshed.

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.

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 routing table” on page 86).

```
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,,STOR

Purpose: 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.

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

Examples: To display TCP/IP storage usage, issue the following:

```
d tcpip,,stor

TCPIP STORAGE
TCPCS    STORAGE  CURRENT  MAXIMUM    LIMIT
TCPCS    ECSA         14M      28M     120M
TCPCS    POOL         52M      62M  NOLIMIT
DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY

or

d tcpip,tcpip2,stor

TCPIP STORAGE
TCPIP2   STORAGE  CURRENT  MAXIMUM    LIMIT
TCPIP2   ECSA      45654K   56823K  204800K
TCPIP2   POOL     124634K  143743K  524288K
DISPLAY TCPIP STOR COMPLETED SUCCESSFULLY
```

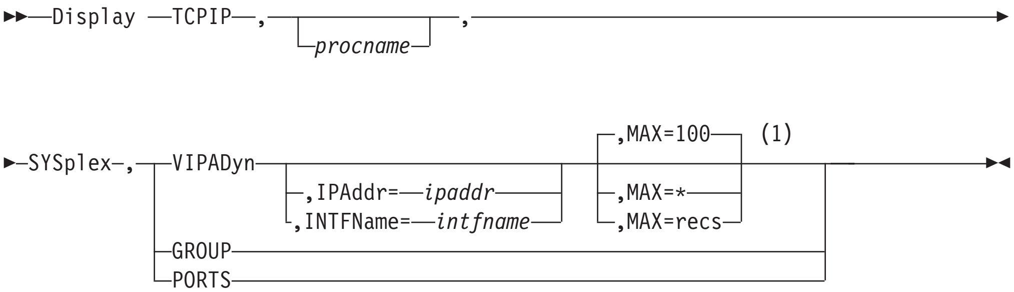
Usage:

- 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.

DISPLAY TCPIP,,SYSPLEX

Purpose: Use the DISPLAY TCPIP,,SYSPLEX command from an operator console to request SYSPLEX information.

Format:



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.

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 range is 1–65 535. 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 or request for LONG format

```
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
```

Usage: See the VIPA information in Virtual IP Addressing in the *z/OS Communications Server: IP Configuration Guide* for an explanation of the fields on the report.

DISPLAY command — TN3270E Telnet server address space

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 3 support the DISPLAY TCPIP command when it is directed to a TN3270E Telnet server.

Table 3. TN3270E Telnet server functions that support the MVS DISPLAY TCPIP command

Function	Command
HELP	“DISPLAY TCPIP,*tnproc*,HELP” on page 94
STOR	“DISPLAY TCPIP,*tnproc*,STOR” on page 96
TELNET	“DISPLAY TCPIP,*tnproc*,TELNET” on page 96

DISPLAY TCPIP,*tnproc*,HELP

Purpose: 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:

```
►►──Display ──TCPIP──,──tnproc──,HElp──┬──────────────────────────┬────────────────────►◄
                                       ├─,STOR────────────────────┤
                                       └─,Telnet──┬──────────────┬┘
                                                  ├─,CLientID────┤
                                                  ├─,CONNection──┤
                                                  ├─,INACTLUS────┤
                                                  ├─,OBJect──────┤
                                                  ├─,PROFile─────┤
                                                  └─,WLM─────────┘
```

Parameters:

STOR
Show help on the Telnet variation of the Display STOR command.

Telnet
Show the available options on the Display Telnet command.

CLientID
Show help on the Display TELNET,CLientID command.

CONNection
Show help on the Display TELNET,CONNection command.

INACTLUS
Show help on the Display TELNET,INACTLUS command.

OBJect
Show help on the Display TELNET,OBJect command.

PROFile
Show help on the Display TELNET,PROFile command.

WLM
Show help on the Display TELNET,WLM command.

Examples: To view the available help for Telnet, issue the following:

```
d tcpip,tnproc,help,Telnet
EZZ6103I D TCPIP,TNPROC,TELNET,
(CLIENTID|OBJECT|PROFILE|CONNECTION|WLM|INACTLUS)
```

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

```
d tcpip,tnproc,help,conn
EZZ6107I D TCPIP,TNPROC,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 TCPIP,*tnproc*,STOR

Purpose: 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.

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.

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. This command does not provide information for the FTP TCP/IP modules.

EZBTNINI
Telnet private storage

EZBTMCTL
Telnet private storage

EZBTPGUE
Telnet private storage

EZBTTMST
Telnet private storage

EZBTZMST
Telnet private storage

Examples: To display Telnet storage usage, issue the following command:

```
d tcpip,Telnet6,stor

EZZ8453I TELNET STORAGE
EZZ8454I TELNET6  STORAGE    CURRENT   MAXIMUM     LIMIT
EZZ8455I TELNET6  ECSA           85K      137K   NOLIMIT
EZZ8455I TELNET6  POOL         6810K     7241K   NOLIMIT
EZZ8455I TELNET6  CTRACE     262372K   262372K   262372K
EZZ8459I DISPLAY TELNET STOR COMPLETED SUCCESSFULLY
```

DISPLAY TCPIP,*tnproc*,TELNET

Purpose: 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.

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 stack is running in IPv6 or the FORMAT LONG configuration statement is specified, then tabular style displays will be used in a format using a second line to display the data when an IP address appears on the line. The displays will be in the single line format if the stack is running in IPv4 and the FORMAT LONG configuration statement is not specified. To ensure uniformity in the displays, if the second line format is in effect, then any IPv4 address is displayed on the second line even if the data would fit on a single line. The tabular displays are:

- D TCPIP,*tnproc*,Telnet,CLientID
- D TCPIP,*tnproc*,Telnet,OBJect
- D TCPIP,*tnproc*,Telnet,CONNection

All commands containing the PROFILE= parameter are considered part of the profile group because the commands categorize (and display) the information based on what profile it is contained in. All of these commands will search all profiles that match the PROFILE= search criteria. Once a match is found, the other parameters will be 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 support multiple dynamic profiles in Telnet. The Display command set includes these categories:

- Profile
- Connection
- Port
- Server

Telnet displays make use of multiple console support (MCS) display lines. In the examples, a C indicates a command line and an L indicates a label line. When MCS is being used, command and label lines do not scroll off the screen.

Tip: All parameters after the command can be in any order.

The following topics provide details on the DISPLAY TCPIP,*tnproc*,TELNET commands that can be used.

DISPLAY Telnet CLientID command: Purpose:

The CLIENTID display can be used to see what Client IDs are defined in the profile and details about the Client ID.

Format:

```
►►──Display TCPIP──,────procname──,────Telnet──,──CLientID──────────────────────►

   ┌─,POrt=ALL────────┐   ┌─,PROFile=CURRent─┐
►──┼──────────────────┼───┼──────────────────┼──┬──────────────────┬───────────►
   ├─,POrt=num────────┤   ├─,PROFile=prfid───┤  ├─,TYPE=clidtype───┤
   ├─,POrt=num1..num2─┤   ├─,PROFile=ACTive──┤  └─,TYPE=WU─────────┘
   └─,POrt=num,qual───┘   ├─,PROFile=ALL─────┤
                          ├─,PROFile=Basic───┤
                          └─,PROFile=Secure──┘
                          ┌─,DETail──┐  ┌─,MAX=100──┐
►──┬──────────────┬──────┼──────────┼──┼───────────┼──────────────────────────────►◄
   └─,ID=clidname─┘      └─,SUMmary─┘  └─,MAX=nn|*─┘
```

Parameters:

procname
: The member name of the cataloged procedure that is used to start the Telnet address space.

Telnet
: Directs the command to the Telnet component.

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** | **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.
- *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
  Produces a listing of Client Identifiers using message EZZ6082I.

  Detail
  Produces a more detailed display showing all Client Identifiers and the Objects mapped to them using message EZZ6081I.

- **TYPE is specified.**

  Summary
  Produces a list of all Client Identifiers for the specified Client Identifier type using message EZZ6082I.

  Detail
  Produces a more detailed display showing all Client Identifiers and the Objects mapped to them for the specified Client Identifier type using message EZZ6081I.

- **ID is specified with or without TYPE.**

  Summary
  Produces a detailed display showing all Client Identifiers and the Objects mapped to them for the specified Client Identifier using message EZZ6081I.

  Detail
  Produces a detailed display showing all Client Identifiers and the Objects mapped to them for the specified Client Identifier using message EZZ6081I. In addition, if the Client Identifier is a group, the individual Client Identifiers within the group are displayed. If a PARMSGROUP is mapped to the Client Identifier, a summary of the resulting parameters used by a connection are displayed.

MAX=100|*nn*|*

The number of data lines displayed. The default is 100. The * (asterisk) means *all*.

Examples:

The following examples show what might be displayed with this command.

```
    D TCPIP,TN3270,TELNET,CLIENTID,PORT=23,PROF=CURR,SUMMARY
(C) EZZ6082I TELNET CLIENTID LIST
    USERID
      USER10
    HOSTNAME
      TESTER12.ANYWHERE.IBM.XXX.YYY.ZZZ.AAA.VVV.CCC.DDD.EEE.
       FFF.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,TN3270,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.ANYWHERE.IBM.XXX.YYY.ZZZ.AAA.BBB.CCC.DDD.EEE.
       FFF.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,TN3270,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)(OATSKTQSWHRT)(DRF)(PCKLECXN2) (IPKPSTS)(SMLT)
      -------  ------------   ---  ---------  -- ----  ----
      *******  **TSBTQ***RT   EC*  BB*******  *P**STS  *DD* *DEFAULT
      LM-----  ------------   DC-  ---------  -------  ---- *TGLOBAL
      ---R---  ---------H--   ---  -B-------  -------  ---- *TPARMS
      -------  ------------   DJ-  ---------  -------  ---- PRMGRP2
      LM*R***  **TSBTQ**HRT   DJ*  BB*******  *P**STS  *DD* <-FINAL
    ----- PORT:    23  ACTIVE                PROF: CURR CONNS:      0
    --------------------------------------------------------------
    28 OF 28 RECORDS DISPLAYED
```

DISPLAY Telnet OBJect command: Purpose:

The OBJECT display can be used to see what Objects are defined in the profile and some details about the object.

Format:

```
►►──DISPLAY TCPIP──,────procname──,────Telnet──,──OBJect──────────────────►

   ┌─,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=Secure──┘

                    ┌─,DETail──┐  ┌─,MAX=100──┐
►──┬──────────────┬─┼──────────┼──┼───────────┼───────────────────────────►◄
   └─,ID=objname──┘ └─,SUMmary─┘  └─,MAX=nn|*─┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

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|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.
- *Basic* is the basic profile.

TYPE=*objtype*
The type of Object Identifier to display. The Object Identifier values are:

- APPLS
- ARAPPL
- DEFAPPL
- DEFAULTS
- INT
- LINEAPPL
- LU
- LUGRP

- LUS
- MAPAPPL
- MONGRP
- PARMSGRP
- PRT
- PRTAPPL
- PRTGRP
- USS
- WU (Determines all 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 will be 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
  Produces a listing of Objects using message EZZ6084I.

  Detail
  Produces a more detailed display showing all Objects and the Client Identifiers to which they are mapped using message EZZ6083I.

- **TYPE is specified.**

  Summary
  Produces a list of all Objects for the specified Object type using message EZZ6084I. Types LUGRP and PRTGRP will 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 inactivated.

  Detail
  Produces a more detailed display showing all Objects and the Client Identifiers to which they are mapped for the specified Object type using message EZZ6083I.

- **ID is specified with or without TYPE.**

  Summary
  Produces a detailed display showing all Objects and the Client Identifiers to which they are mapped for the specified Object using message EZZ6083I.

  Detail
  Produces a detailed display showing all Objects and the Client Identifiers to which they are mapped for the specified Object using message EZZ6083I. In addition, if the Object is a group, the individual Objects within the group are displayed.

MAX=100 | *nn* | *

The number of data lines displayed. The default is 100. The asterisk (*) means *all*.

Examples:

The following examples show what might be displayed with this command.

```
    D TCPIP,TN3270,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,TN3270,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,TN3270,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: **Purpose:**

The PROFILE DISPLAY command allows the system administrator to determine:

- What 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──,────procname──,────Telnet──,──PROFile──────────────────────────►

   ┌─,POrt=ALL────────┐   ┌─,PROFile=CURRent─┐  ┌─SUMmary─┐  ┌─,MAX=100──┐
►──┼──────────────────┼───┼──────────────────┼──┼─────────┼──┼───────────┼──────────►◄
   ├─,POrt=num────────┤   ├─,PROFile=prfid───┤  └─DETail──┘  └─,MAX=nn|*─┘
   ├─,POrt=num1..num2─┤   ├─,PROFile=ACTive──┤
   └─,POrt=num,qual───┘   ├─,PROFile=ALL─────┤
                          ├─,PROFile=Basic───┤
                          └─,PROFile=Secure──┘
```

Parameters:

procname
: The member name of the cataloged procedure used to start the Telnet address space.

Telnet
: Directs the command to the Telnet component.

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|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.
- *Basic* is the basic profile.

SUMmary|DETail
: SUMmary displays the total number of users associated with the profile, selected options, and SMF record subtype. DETail displays SSL information, timeout values, and maximum limits.

MAX=100|*nn***|***
: The number of data lines displayed. The default is 100. The * (asterisk) means *all.*

Examples:

```
    D TCPIP,TN3270,TELNET,PROF,PORT=23
(C) EZZ6060I TELNET PROFILE DISPLAY
(L)   PERSIS   FUNCTION      DIA  SECURITY    TIMERS  MISC
(L)  (LMTGCAK)(OATSKTQSWHRT)(DRF)(PCKLECXN2)(IPKPSTS)(SMLT)
      -------- ------------  ---  ---------  -------  ----
      LM*R*P*  **TSBTQ*WHRT  DJ*  BB*******  *P**STS  SDD*
    ----- PORT:       23    ACTIVE           PROF: CURR CONNS:      0
    ------------------------------------------------------------
       FORMAT            LONG
       TNSACONFIG        DISABLED
    5 OF 5 RECORDS DISPLAYED


    D TCPIP,TN3270,TELNET,PROF,PORT=23,DETAIL
(C) EZZ6080I TELNET PROFILE DISPLAY
(L)  PERSIS   FUNCTION      DIA  SECURITY   TIMERS  MISC
(L) (LMTGCAK)(OATSKTQSWHRT)(DRF)(PCKLECXN2)(IPKPSTS)(SMLT)
     -------  ------------  ---  ---------  -------  ----
     *******  **TSBTQ***RT  EC*  BB*******  *P**STS  *DD* *DEFAULT
     -------  ----BT--WHRT  DJ-  ---L---*-  -------  S--- *TGLOBAL
     LM-R-P-  ----BT--WHRT  ---  -B-*-----  ----ST-  ---- *TPARMS
     LM*R*P*  **TSBTQ*WHRT  DJ*  BB*******  *P**STS  SDD* CURR
    PERSISTENCE
      LUSESSIONPEND
      MSG07
      NO TKOSPECLU
      TKOGENLURECON           2 NOKEEPONTMRESET
      NOCHECKCLIENTCONN
      DROPASSOCPRINTER
      KEEPLU                  0 (OFF)
    FUNCTIONS
      NOOLDSOLICITOR
      NOSINGLEATTN
      TN3270E
      SNAEXTENT
      UNLOCKKEYBOARD BEFOREREAD
      UNLOCKKEYBOARD TN3270BIND
      SEQUENTIALLU
      NOSIMCLIENTLU
      WLMCLUSTERNAME
      HNLOOKUP
      REFRESHMSG10
      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             DETAIL
      DEBUG ROUTING     JOBLOG
      NOFULLDATATRACE
    SECURITY
      BASICPORT
      CONNTYPE          BASIC
      KEYRING           NONE
      CRLLDAPSERVER     NONE
      ENCRYPTION        NONE
      CLIENTAUTH        NONE
      NOEXPRESSLOGON
      NONACUSERID
      NOSSLV2
    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
        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
   TCPIPJOBNAME     NO AFFINITY
   TNSACONFIG       DISABLED
92 OF 92 RECORDS DISPLAYED
```

DISPLAY TELNET CONNECTION command: Purpose:

The CONNECTION DISPLAY command with the SUMmary parameter specified allows the system administrator to get a high-level view of what connections exist and what they are being used for.

The DISPLAY command with the DETail parameter specified gives the system administrator a complete look at one connection. It will show all of the information available regarding a single connection.

Format:

```
►►── DISPLAY TCPIP──,────procname──,────Telnet──,──CONNection────────────────────►
```

```
   ┌─,POrt=ALL──────────┐   ┌─,PROFile=ALL──────┐
►──┼────────────────────┼───┼───────────────────┼──────────────────────────────────────►
   ├─,POrt=num──────────┤   ├─,PROFile=prfid────┤
   ├─,POrt=num1..num2───┤   ├─,PROFile=ACTive───┤
   └─,POrt=num,qual─────┘   ├─,PROFile=CURRent──┤
                            ├─,PROFile=Basic────┤
                            └─,PROFile=Secure───┘

                                                                      ┌─,MAX=100──┐
►──┬──────────────────────────────────────────────────────────────┬───┼───────────┼──►◄
   │                                   ┌─,DETail──┐               │   └─,MAX=nn|*─┘
   ├──┬─,COnn=connid─────────┬─────────┼──────────┼───────────────┤
   │  ├─,IPPort=ipaddr..port─┤         └─,SUMmary─┘               │
   │  └─,LUName=luname───────┘                                    │
   │                                               ┌─,NOHname─┐   │
   ├──┬─,LUName=luname*──────────────────────┬─────┼──────────┼───┤
   │  ├─,APPL=applname|applname*─────────────┤     └─,HName───┘   │
   │  ├─,TCPipjobname=tcpip──────────────────┤                    │
   │  ├─,IPAddr=──┬─ipaddr───────────────┬───┤                    │
   │  │           ├─ipv4mask:ipv4subnet──┤   │                    │
   │  │           └─ipv6addr/prefixlen───┘   │                    │
   │  ├─,LUGroup=lugroupname─────────────────┤                    │
   │  ├─,IPGroup=ipgroupname─────────────────┤                    │
   │  └─,PROTOcol=protocol mode──────────────┘                    │
   │                                   ┌─,HName───┐               │
   └──┬─,HName=*hostname─────┬─────────┼──────────┼───────────────┘
      └─,HNGroup=hngroupname─┘         └─,NOHname─┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

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. This filter applies only when Telnet is running in its own address space.

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 data lines displayed. The default is 100. The * (asterisk) means *all*.

Examples:

```
    D TCPIP,TN3270,TELNET,CONN
(C) EZZ6064I TELNET CONNECTION DISPLAY
(L)          EN                                        TSP
(L) CONN     TY 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 SSL-related information.

```
D TCPIP,TELNET3,T,CONN,CONN=35
EZZ6065I TELNET CONNECTION 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  4S 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)(OATSKTQSWHRT)(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*WHRT  DJ*  TSS*DF***  *P**STS  SDD* TP-CURR
        PARMSGROUP: IG IPGRP1
      -------  O-----------  TC-  --------2  I------  ---- PRMGRP1
        LUMAP-PMAP: ON LUMAP  OF LUGRP1
      -------  -----------   --F  ---------  -------  -M-- PRMGRP2
      LM*R*P*  O*TSBTQ*WHRT  TCF  TSS*DF**2  IP**STS  SMD* <-FINAL
    56 OF 56 RECORDS DISPLAYED
```

Usage:

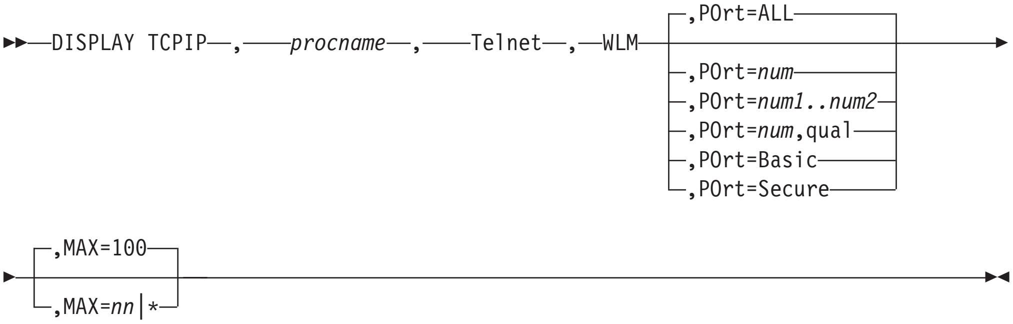
Only one connection at a time will be displayed with parameters CONN=, IPPort=, and LUName= if no wildcard is on LUName.

DISPLAY Telnet WLM command: Purpose:

The WLM DISPLAY command allows the system administrator to determine what names Telnet has used to register itself with the Workload Manager (WLM) and which of these registered names is available to users. These names are available to clients only when DNS is configured for Connection Optimization. See Connection optimization in a sysplex domain in the *z/OS Communications Server: IP Configuration Guide* for more information about connection optimization.

Because TELNET in a QUIESCE state will unregister itself, the system administrator can also use this command to determine whether TELNET is in a QUIESCE or STARTed state.

Format:



Parameters:

procname
: The member name of the cataloged procedure used to start the Telnet address space.

Telnet
: Directs the command to the Telnet component.

WLM
: The workload manager 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.

MAX=100|*nn*|*
The number of data lines displayed. The default is 100. The * (asterisk) means *all*.

Examples:

```
D TCPIP,,T,WLM

(C) EZZ6067I TELNET WLM DISPLAY
(L) WLM CLUSTER NAME          STATUS
    -----------------  ---------------------
    WLM2               REGISTERED
    WLM1               REGISTERED
    2 OF 2 RECORDS DISPLAYED
```

DISPLAY Telnet INACTLUS command: Purpose:

The INACTLUS DISPLAY command allows a system administrator to see all of the LUs that are not available to any users since the VARY INACT command was issued or the OPEN ACB failed and Telnet automatically sets the LU inactive.

Format:

```
                                                         ┌,MAX=100───┐
►►──DISPLAY TCPIP──,────procname──,────Telnet──,──INACTLUS──┼───────────┼───────────►◄
                                                         └,MAX=nn|*──┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

INACTLUS
The inactive LUs keyword.

MAX=100|*nn*|*
The number of data lines displayed. The default is 100. The * (asterisk) means *all*.

Examples:

```
D TCPIP,,T,INACTLUS

(C) EZZ6061I TELNET INACTLUS DISPLAY
(L) INACTIVE LUS
                   TCPM1003  TCPM1005  TCPM1004  TCPM1001  TCPM1010
                   TCPM1015  TCPM1012  TCPM1008
    2 OF 2 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 4.

Table 4. Servers or address spaces that support the MVS Modify command

Server/Addr space	Main parameters	Additional information
Automated domain name registration application (EZBADNR)	DEBUG, DISPLAY, REFRESH	"MODIFY command—automated domain name registration application (EZBADNR)" on page 114
FTP server	DUMP, DEBUG	"MODIFY command—FTP" on page 132
IKE server	DISPLAY, REFRESH	"MODIFY command—IKE server" on page 138
Load Balancing Advisor	DEBUG, DISPLAY	"MODIFY command—z/OS Load Balancing Advisor" on page 178
Load Balancing Agent	DEBUG, DISPLAY, QUIESCE, ENABLE	"MODIFY command—z/OS Load Balancing Agent" on page 186
NCPROUTE server	C, PARMS, PROFILE, QUERY, GATEWAYS, TABLES	"MODIFY command—NCPROUTE" on page 139
Network security services server	DISPLAY, REFRESH	"MODIFY command—network security services server" on page 142
OMPROUTE	KILL, RECONFIG, ROUTESA, OSPF, RIP, GENERIC, RTTABLE, IPV6OSPF, IPV6RIP, GENERIC6, RT6TABLE, TRACE, DEBUG, TRACE6, DEBUG6, SADEBUG	"MODIFY command—OMPROUTE" on page 143
Policy Agent	LOGLEVEL, TRACE, DEBUG, QUERY, REFRESH, MEMTRC, UPDATE	"MODIFY command—Policy Agent" on page 155

Table 4. Servers or address spaces that support the MVS Modify command (continued)

Server/Addr space	Main parameters	Additional information
Resolver Address Space	DISPLAY, REFRESH	"MODIFY command—Resolver address space" on page 157
REXEC	EXIT, TSOPROC, MSGCLASS, TSCLASS, TRACE,PURGE	"MODIFY command—REXEC" on page 159
Rpcbind server	TRACE	"MODIFY command—RPCBIND" on page 160
SMTP	SMSG	"MODIFY command—SMTP" on page 161
SNALINK LU0	HALT	"MODIFY command—SNALINK LU0" on page 167
SNALINK LU6.2	CANCEL, DROP, HALT, LIST, RESTART, TRACE	"MODIFY command—SNALINK LU 6.2" on page 168
SNMP Agent	INTERVAL, TRACE	"MODIFY command—SNMP agent" on page 172
SNMP Network SLAPM2 subagent	DEBUG, CACHE, QUERY	"MODIFY command—SNMP Network SLAPM2 subagent" on page 173
TNF	DISPLAY, REMOVE	"MODIFY command—VMCF and TNF" on page 175
Trap Forwarder Daemon	QUERY, REFRESH, TRACE	"MODIFY command—Trap forwarder daemon (TRAPFWD)" on page 174
VMCF	DISPLAY, REMOVE	"MODIFY command—VMCF and TNF" on page 175
X.25 NPSI server	CANCEL, DEBUG, EVENTS, HALT, LIST, RESTART, SNAP, TRACE, TRAFFIC	"MODIFY command—X.25 NPSI server" on page 176

MODIFY command—automated domain name registration application (EZBADNR)

Purpose

Use the MODIFY command to control the automated domain name registration (ADNR) application from the operator's console.

Format

```
►►──┬─MODIFY─┬──procname,──────────────────────────────────────────────────────────────────────────────────►
    └─F──────┘

►──┬─DEBug,Level=debug_level───────────────────────────────────────────────────────────────────────────────┬───►◄
   ├─DISplay,──┬─DEBug───────────────────────────────────────────────────────────────────────────────────┬─┤
   │           │                                                              ┌─,SUMMARY─┐ ┌─,MAX=100──┐ │ │
   │           ├─DNS──┬──────────────────┬─┬────────────────────────────────┬─┼──────────┼─┼───────────┼─┤ │
   │           │      └─,DNSID=dns_label─┘ └─,ZONES──┬────────────────────┬─┘ └─,DETAIL──┘ ├─,MAX=*────┤ │ │
   │           │                                     └─,ZONEID=zone_label─┘                └─,MAX=recs─┘ │ │
   │           │                                            ┌─,SUMMARY─┐ ┌─,MAX=100──┐                   │ │
   │           └─GWM──┬───────────────────────────────────┬─┼──────────┼─┼───────────┼───────────────────┘ │
   │                  └─,GROUPS──┬──────────────────────┬─┘ └─,DETAIL──┘ ├─,MAX=*────┤                     │
   │                             └─,GROUPID=group_label─┘                └─,MAX=recs─┘                     │
   └─REFRESH───────────────────────────────────────────────────────────────────────────────────────────────┘
```

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. Summary DNS information includes the following:

- 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. Detailed DNS information includes the following:

- 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.

Summary zone information includes the following:

- DNS label
- DNS status
- Zone information

For each zone the following 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.

Detailed zone information includes the following:

- DNS label
- DNS status
- DNS IP address and port
- Number of zones defined
- Number of zones active
- Zone information

For each zone, the following is displayed:

- Zone label
- Zone status
- Status timestamp
- Domain suffix
- TSIG flags
- DNS resource record information

For each DNS resource record, the following 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. Summary GWM information includes the following:

- GWM label
- GWM status

DISplay,GWM,DETAIL

Displays detailed GWM information. Detailed GWM information includes the following:

- 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. Summary group information includes the following:

- GWM label
- GWM status
- Group information

For each group, the following 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.

Detailed group information includes the following:

- 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 is displayed:

- Group label
- Group name
- Group type
- DNS label
- Zone label
- Member information

For each member, the following is displayed:

- Member *hostname* (if available)
- IP address information

For each member IP address, the following 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.

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 are 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 are 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 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 are 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 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 are 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 are 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—FTP

Purpose

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.

When migrating from a release prior to z/OS V1R2, the MODIFY commands are mapped as follows:

Table 5. MODIFY command mapping.

Releases prior to z/OS V1R2	Current z/OS release
`modify jobname,TRACE`	`modify jobname,DEBUG=(BAS)`
`modify jobname,NOTRACE`	`modify jobname,DEBUG=(NONE)`
`modify jobname,JTRACE`	`modify jobname,DEBUG=(CMD,FSC,JES)`
`modify jobname,NOJTRACE`	`modify jobname,DEBUG=(NONE)`
`modify jobname,DUMP`	`not supported`
`modify jobname,NODUMP`	`modify jobname,DUMP=(NONE)`
`modify jobname,JDUMP`	`not supported`
`modify jobname,NOJDUMP`	`not supported`
`modify jobname,UTRACE`	`not supported`
`modify jobname,NOUTRACE`	`not supported`

Format

```
                                             ┌─────────────────┐
                                             ▼                 │
►►──┬─MODIFY─┬──jobname──,──┬─DEBug─=─(─┬──────────┬─,──┴──┬────────────────────►
    └─F──────┘              │           ├─?────────┤       │
                            │           ├─ACC──────┤       │
                            │           ├─ALL──────┤       │
                            │           ├─BAS──────┤       │
                            │           ├─CMD──────┤       │
                            │           ├─FLO──────┤       │
                            │           ├─FSC(n)───┤       │
                            │           ├─INT──────┤       │
                            │           ├─JES──────┤       │
                            │           ├─NONE─────┤       │
                            │           ├─PAR──────┤       │
                            │           ├─SEC──────┤       │
                            │           ├─SOC(n)───┤       │
                            │           ├─SQL──────┤       │
                            │           ├─UTL──────┤       │
                            │           │     (1)  │       │
                            │           └─Xyyy─────┘       │
                            │           ┌────────────────┐ │
                            │           ▼                │ │
                            └─DUmp─=─(─┬──────────┬─,──┴─┘
                                       ├─?────────┤
                                       ├─n────────┤
                                       ├─ALL──────┤
                                       ├─FSC──────┤
                                       ├─JES──────┤
                                       ├─NONE─────┤
                                       ├─SOC──────┤
                                       ├─SQL──────┤
                                       │     (1)  │
                                       └─Xyyy─────┘

►──┬──────────────────────┬─)──────────────────────────────────────────────►◄
   ├─USERID(filter_name)──┤
   └─IPADDR(filter)───────┘
```

Notes:

1 Prepend any option *yyy* with X to turn off that trace.

Parameters

DEBug

Subcommand to begin general tracing. Options for general tracing include the following:

? 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. Options for extended tracing include the following:

? 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 will be set to the current value of the options. Otherwise, there will be 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 will activate 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:

  ```
  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))
  ```

  The use of the ALL parameter can produce an extensive amount of trace data and should not be specified on a routine basis.

- 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

Purpose

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 used to start the IKE server (IKED).

DISPLAY
Displays configuration values 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 on which parameters can be dynamically changed. See IkeConfig in the *z/OS Communications Server: IP Configuration Reference* for more information.

FILE
Indicates the name and location of the IKE server configuration file to be read. The *filename* value must be a fully qualified z/OS UNIX file name or an MVS data set specification. 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 file name in single quotation marks. The default value is /etc/security/iked.conf. This option is valid only when specified with REFRESH.

MODIFY command—NCPROUTE

Purpose

You can control most of the functions of the NCPROUTE address space from the operator console using the MODIFY command. Following is 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 6 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 6. 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

Table 6. NCPROUTE Modify parameters (continued)

Parameter	NCPROUTE GATEWAY option	Host routes	Network routes	Advertise as default router	Local routes	Unreachable routes
-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

Purpose

You can use the operator console and the MODIFY command to control the network security services (NSS) 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 network security services server daemon (NSSD).

DISPLAY
    Displays configuration values that are currently being used by the NSS server.

REFRESH
    Indicates that the NSS server configuration file should be reread. Not all NSS server parameters can be updated using this command. See the description for the parameters in the configuration file to find out which ones can be dynamically changed. See the Network security services server information in the *z/OS Communications Server: IP Configuration Reference* for more information.

FILE
    Indicates the name and location of the (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 specification. 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 file name in single quotation marks. The default value is /etc/security/nssd.conf. This option is valid only when specified with REFRESH.

MODIFY command—OMPROUTE

Purpose

You can control OMPROUTE from the operator console using the MODIFY command.

Format

```
►►──┬─MODIFY─┬──procname──,──────────────────────────────────────────────────►
    └─F──────┘

►──┬─KILL─────────────────────────────────────────────────┬──────────────────►◄
   ├─RECONFIG─────────────────────────────────────────────┤
   ├─ROUTESA=──┬─ENABLE──┬────────────────────────────────┤
   │           └─DISABLE─┘                                │
   ├─TRACE=trace_level────────────────────────────────────┤
   ├─DEBUG=debug_level────────────────────────────────────┤
   ├─TRACE6=trace6_level──────────────────────────────────┤
   ├─DEBUG6=debug6_level──────────────────────────────────┤
   ├─SADEBUG=sadebug_level────────────────────────────────┤
   ├─OSPF─┤ OSPF options ├────────────────────────────────┤
   ├─RIP─┤ RIP options ├──────────────────────────────────┤
   ├─GENERIC─┤ GENERIC options ├──────────────────────────┤
   ├─RTTABLE──┬──────────────────────────┬─┬────────────┬─┤
   │          └─,PRtable=──┬─ALL────┬─┘ ├─,DEST=ip_addr─┤ │
   │                       └─prname─┘   └─,DELETED──────┘ │
   ├─IPV6OSPF─┤ IPv6 OSPF options ├───────────────────────┤
   ├─IPV6RIP─┤ IPv6 RIP options ├─────────────────────────┤
   ├─GENERIC6─┤ GENERIC6 options ├────────────────────────┤
   └─RT6TABLE──┬──────────────────────────────────┬───────┘
               ├─,DEST=──┬─ip_addr───────────┬────┤
               │         └─ip_addr/prefixlen─┘    │
               └─,DELETED─────────────────────────┘
```

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─────────────┤
   └─,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──────┤          │
   │             └─,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───────────────────────────────────┤
   └─,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.

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: These new configuration statements must be reread from the configuration file through this command before the interface is configured to the TCP/IP stack.

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 use for initialization and IPv4 routing protocols.

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.

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.

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 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 route 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 route tables appear in the display of OMPROUTE route tables. A policy-based route table is active if it is referenced by an active routing rule and its associated action.
- This option displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The contents of the OMPROUTE routing table might contain information that is different from that 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 7.

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 all the routes in the OMPROUTE IPv6 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.

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.
- This option displays the contents of the working table that is used by OMPROUTE; it does not display the TCP/IP routing table. The contents of the OMPROUTE routing table might contain information that is different from that 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 7.

Examples

You can use MODIFY OMPROUTE commands to perform functions that include the following:

- "Displaying OMPROUTE information"
- "Stopping OMPROUTE"
- "Rereading the configuration file"
- "Enabling or disabling the OMPROUTE subagent" on page 154
- "Changing the cost of OSPF links" on page 154

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 18 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 more information, see ROUTESA_CONFIG 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

Purpose

You can use the operator console and the MODIFY command to control the Policy Agent functions.

Format

```
►►──┬─MODIFY─┬──procname──,──┬─LOGLEVEL──,──LEVEL=n─┬──────────────────►◄
    └─F──────┘               ├─TRACE──,──LEVEL=t────┤
                             ├─DEBUG──,──LEVEL=d────┤
                             ├─MEMTRC───────────────┤
                             ├─QUERY────────────────┤
                             ├─REFRESH──────────────┤
                             └─UPDATE───────────────┘
```

Parameters

procname
: The member name of the cataloged procedure used to start the Policy Agent.

LOGLEVEL,LEVEL=*n*
: Changes the Policy Agent LogLevel. The desired log level is *n*. If *n* is not specified, then the current LogLevel remains the same. See LogLevel statement 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 desired trace level is *t*. If *t* is not specified, then the current trace level remains the same. See Starting Policy Agent from the z/OS shell 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 desired 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 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 information in the *z/OS Communications Server: IP Configuration Guide* for more information concerning the FLUSH/NOFLUSH and PURGE/NOPURGE parameters.

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.

MODIFY command—Resolver address space

Purpose

You can refresh the Resolver Address Space from the operator console using the MODIFY command. The REFRESH command allows you to refresh the Resolver Address Space and the DISPLAY command allows you to display the current values of the resolver setup statements.

For a description of the resolver setup statements, see Resolver setup statements in the *z/OS Communications Server: IP Configuration Reference*.

Format

```
►►──┬─MODIFY─┬──procname──,──┬───────────────────────────────────┬───────►◄
    └─F──────┘               ├─Display───────────────────────────┤
                             └─REFRESH──┬──────────────────────┬─┘
                                        └─,SETUP=──┬─xxx──────┬┘
                                                   ├─xxx(yyy)─┤
                                                   └─'/xxx'───┘
```

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 will include a line as follows:

```
RESOLVER PROC   = DEFAULT
```

  If `DEFAULT` is displayed, then the *procname* value is `RESOLVER`. Any other value should be used as the *procname*.

Display
: Displays the current resolver setup statement values.

REFRESH
: Causes applications to have their TCPIP.DATA information, including local host tables (for example, etc/hosts, etc/ipnodes, HOSTS.SITEINFO, HOSTS.ADDRINFO, or ETC.IPNODES information), updated on their next Resolver request after the REFRESH occurs.

  SETUP=
  : The contents of the specified resolver setup file are processed. As previously described with the REFRESH parameter, processing TCPIP.DATA statements and local host tables are updated at the next resolver request.

    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.

Note: 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.

```
f resolver,display

EZZ9298I DEFAULTTCPIPDATA - None
EZZ9298I GLOBALTCPIPDATA - SYS1.TCPPARMS(TCPDATA)
EZZ9298I DEFAULTIPNODES - USER55.ETC.IPNODES
EZZ9298I GLOBALIPNODES - None
EZZ9304I NOCOMMONSEARCH
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
EZZ9293I REFRESH COMMAND PROCESSED
```

Usage

See the Resolver setup statements in the Customization information in the *z/OS Communications Server: IP Configuration Guide* for an explanation of the fields on the report.

MODIFY command—REXEC

Purpose

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

Purpose

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:

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

Purpose

The MODIFY SMTP command provides an interactive interface to the SMTP server that allows you to do the following:

- 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───────────────────┬──────────────────►◄
                                   ├─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 regarding 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 regarding 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's console for the QUeues report. Valid values are in the range 1-65 535. An asterisk (*) causes all output lines up to line 65 535 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 retrying the query. The RETRYAGE statement specifies the number of days SMTP should continue 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 Internet 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:

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

Purpose

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.

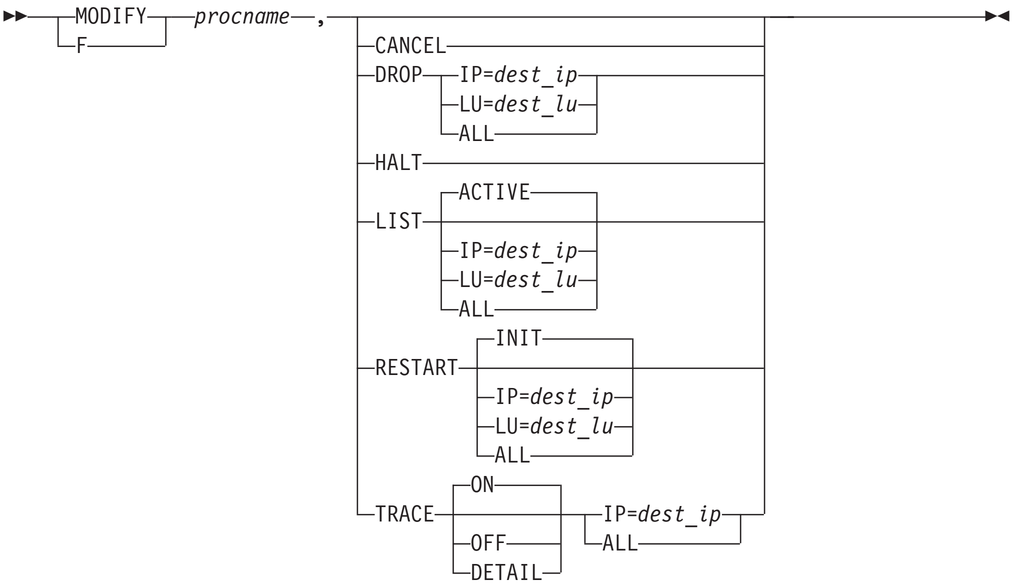
MODIFY command—SNALINK LU 6.2

Purpose

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



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 retries the connection 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

Purpose

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

Purpose

You can control the Network SLAPM2 subagent (nslapm2) functions from the operator console using the MODIFY command. The following is 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 desired 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 desired 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—Trap forwarder daemon (TRAPFWD)

Purpose

You can control the TRAPFWD daemon from the operations console using the MODIFY command. The following is 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

Purpose

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

Purpose

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

```
AUTOLOG
   TCPIPX25
```

you could issue either of these commands at the operator console:

```
MODIFY TCPIPX25,HALT

F TCPIPX25,HALT
```

MODIFY command—z/OS Load Balancing Advisor

Purpose

You can control the z/OS Load Balancing Advisor from the operator's 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 desired 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 on 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" on page 179.

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 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 will show 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 will show 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 may 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

Purpose

You can control the z/OS Load Balancing Agent from the operator's console using the MODIFY command.

Format

```
├──┬─MODIFY─┬──procname,──────────────────────────────────────────────────────┤
   └─F──────┘


├──┬─DEBug,Level=debuglevel─────────────────────────────────────────┬─────────┤
   ├─DISplay,──┬─DEBug──────────────────────────────────────────────┤
   │           │                                  ┌─,MAX=100──┐   │
   │           └──┬─MEMbers─────────────────────┬─┼───────────┼───┘
   │              ├─MEMbers,DETail──────────────┤ ├─,MAX=*────┤
   │              ├─MEMbers,DETail,PORT=portnum─┤ └─,MAX=recs─┘
   │              └─MEMbers,DETail,TCPname=tcpname─┘
   ├─Enable,Target options───────────────────────────────────────────┤
   └─Quiesce,Target options──────────────────────────────────────────┘
```

Target options:

```
                         ┌─,PROTOcol=TCP───┐
├──┬─PORT=portnum────────┼─────────────────┼──┬──────────────────┬─┬───────────┤
   │                     └─,PROTOcol=proto─┘  └─,IPaddr=ipaddr───┘ │
   ├─TCPname=tcpname───────────────────────────────────────────────┤
   └─SYStem────────────────────────────────────────────────────────┘
```

Parameters

procname
The member name of the cataloged procedure used to start the Agent.

DEBug,Level=*debuglevel*
Changes the Agent debug level. The desired 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 on 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 will advise the load balancer to route work to the target applications.

Quiesce
Mark all matching active registered applications or system members as quiesced. The Agent will advise 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 specifying 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" on page 179.

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).

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 will calculate 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 7.

Table 7. 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, SYSPLEX	See "VARY command — TCP/IP address space" on page 191
TN3270E Telnet server address space	HELP, OBEYFILE, TELNET	See "VARY command — TN3270E Telnet server address space" on page 218

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 8 support the VARY TCPIP command when it is directed to a TCP/IP stack address space.

Table 8. Functions that support the VARY TCPIP command.

Function	Command
DATTRACE	“VARY TCPIP,,DATTRACE” on page 191
DROP	“VARY TCPIP,,DROP” on page 193
OBEYFILE	“VARY TCPIP,,OBEYFILE” on page 194
OSAENTA	“VARY TCPIP,,OSAENTA” on page 195
PKTTRACE	“VARY TCPIP,,PKTTRACE” on page 204
PURGECACHE	“VARY TCPIP,,PURGECACHE” on page 208
START or STOP	“VARY TCPIP,,START or VARY TCPIP,,STOP” on page 209
SYSPLEX	“VARY TCPIP,,SYSPLEX” on page 210

VARY TCPIP,,DATTRACE

Purpose: Use the VARY TCPIP,,DATTRACE command to trace socket data (transforms) into and out of the physical file structure (PFS).

Format:

```
►►──Vary──TCPIP,──┬──────────┬──,──DATtrace──┬───────┬──┤ TRACE ├──────────────────►◄
                  └─procname─┘               ├─,ON───┤
                                             └─,OFF──┘
```

TRACE:

```
   ┌─FULL──────────────────────────────┐      ┌─JOBNAME=*────────────┐
├──┼───────────────────────────────────┼──,───┼──────────────────────┼──,──────────────►
   │                ┌─200──────────────┐│      └─JOBNAME=job_name─────┘
   └─ABBREV=────────┴─abbrev_length────┴┘

   ┌─IP=*─────────────────────────────┐
►──┼──────────────────────────────────┼──────────────────────────────────────────────┤
   └─IP=──┬─┤ IPv4_address ├─┬────────┘
          └─┤ 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 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.

FULL
Specifies that the entire IP packet 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 will be 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 will be traced. If the IP option is omitted, or an asterisk (*) is specified, then all IP addresses will be 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 will be displayed if both are coded.

Note: IP address selection is not recommended for use with DATTRACE.

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
```

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

 JobName: *          TrRecCnt: 00000000  Length: FULL
 IpAddr/PrefixLen:  5555:4444::2222/128
```

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

Purpose: Use the VARY TCPIP,,DROP command to drop a connection. For detailed information about drop processing, see “Netstat DRop/-D command” on page 370.

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 required after specifying the DRop parameter. Synonymous syntax parameter to select the connection identifier (*connid*) for the TCP/IP socket connection that is to be dropped. 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.

Examples: Following are examples of 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
  ```

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.DROP.

VARY TCPIP,,OBEYFILE

Purpose: Use the VARY TCPIP,,OBEYFILE command to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCP/IP address space.

See the *z/OS Communications Server: IP Configuration Guide* 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 containing TCP/IP 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: Following are examples of 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 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 configuration statements:

  ```
  VARY TCPIP,,O,USER99.TCPIP(OBEYFIL2)
  ```

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.OBEYFILE.
- The DSN= parameter cannot be a z/OS UNIX file.

VARY TCPIP,,OSAENTA

Purpose: 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.

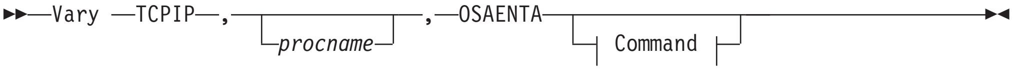
The OSAENTA command dynamically defines a QDIO interface to the OSA-Express adapter being traced, called an OSAENTA interface. That interface is used exclusively for capturing OSA-Express network traffic analyzer traces.

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 containing 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.

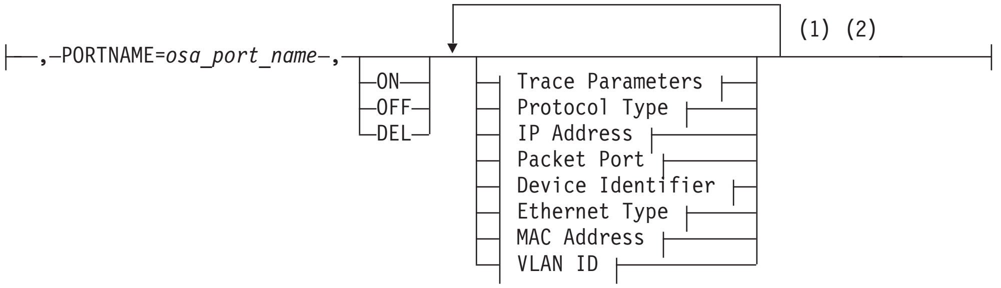
Tips:

- 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.

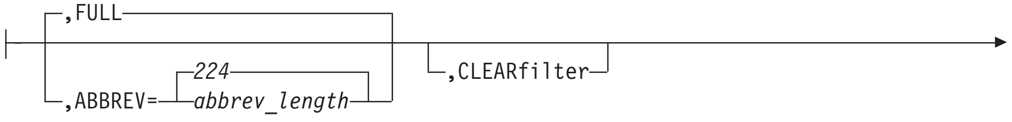
Format:



Command:



Trace Parameters:



```
|                                    ┌─,DISCARD=EXCEPTION────┐
  ►──┬──────────────────────────┬────┼───────────────────────┼──────────────►
     │         ┌─1024──────────┐│    │                       │
     └─,DATA=──┴─trace_amount──┴┘    ├─,DISCARD=ALL──────────┤
                                     ├─,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───────┘
```

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. Refer to *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 desired. This is the same port name that is defined on the VTAM TRLE statement PORTNAME keyword. This parameter is required.

Tip: You are not required to also define OSA-Express to TCP/IP using the DEVICE/LINK or INTERFACE statement in order to collect trace data.

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 – 65 472 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.

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 inactivate 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 data device that is active on a z/OS stack, you can obtain the *device_id* value for that data 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. The following are 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. This option should be used 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 desired 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 2 147 483 647 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 containing 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 twelve 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 very 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 – 65 535. 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.

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. Since 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

Purpose: 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 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──┴─────────────────────────────────────┤
```

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. Refer to *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*
LINKName specifies the name of the link (*link_name*) defined in the preceding LINK statement. INTFName specifies the interface name (*intf_name*). If the LINKName/INTFName parameter is omitted or an asterisk (*) is specified for either parameter, the PKTTRACE parameters will 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.

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.

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 multiple PKTTRACE commands are issued. Use the NETSTAT DEvlinks (**netstat -d**) command to display the results. An IP packet is traced according to the first setting that matches.
- 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

Purpose: Use the VARY TCPIP,,PURGECACHE command to delete the ARP cache entries for a link or to delete neighbor cache entries for an interface.

Format:

```
►►──Vary ──TCPIP──,──┬──────────┬──,──PURGECache,name──────────────────────────►◄
                     └─procname─┘
```

Parameters:

name

The interface name or link name of the cache to be purged.

If the *name* matches a link name, the local ARP cache or the outboard OSA cache entries (for QDIO token ring and QDIO Ethernet) for that link is purged. If the *name* matches an interface name, the IPv6 neighbor cache for that interface is purged.

Notes:

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 link and the OSA-Express is shared by multiple stacks, then this command will purge the ARP cache for all stacks which share the OSA (because OSA-Express maintains a single ARP cache for all stacks which share it).
2. Translate entries are not deleted for ATM or LCS links. For ATM:
   - PVC and ATMARP server entries are not deleted.
   - ACTIVE SVC entries are not deleted because TCPIP processing periodically validates these entries.
   - A clear might be needed for SVC entries that are not ACTIVE. When the asynchronous clear completes, the entries will be deleted.

Examples: Following is an example of using PURGECache.

- From TSO:

```
 netstat arp all
MVS TCP/IP NETSTAT CS V1R9 TCPIP Name: TCPCS
 Querying ARP cache for address 9.67.113.1
```

```
Link: TR1 IBMTR: 000BC6AA1B88
Route info: 0000

Querying ARP cache for address 9.67.113.61
Link: 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
Link: 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

Purpose: 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: Following is an example of starting 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,,SYSPLEX

Purpose: Use the VARY TCPIP,,SYSPLEX command to change the TCP/IP stack's sysplex configuration.

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 inactivate 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: This should be done 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.

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,REACTIVE 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 9 support the VARY TCPIP command when it is directed to a TN3270E Telnet server.

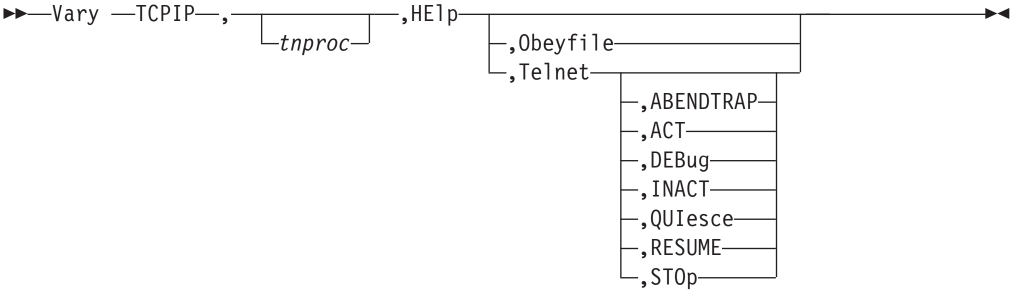
Table 9. TN3270E Telnet servers that support the MVS VARY TCPIP command

Function	Command
HELP	“VARY TCPIP,*tnproc*,HELP”
OBEYFILE	“VARY TCPIP,*tnproc*,OBEYFILE” on page 219
TELNET	“VARY TCPIP,*tnproc*,TELNET” on page 220

VARY TCPIP,*tnproc*,HELP

Purpose: 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:



Parameters:

Obeyfile
Show help on the VARY OBEYFILE command.

Telnet
Show the available options on the DISPLAY TELNET command.

ABENDTRAP
Show help on the VARY TELNET,ABENDTRAP command.

ACT
Show help on the VARY TELNET,ACT command.

DEBug
Show help on the VARY TELNET,DEBUG command.

INACT
Show help on the VARY TELNET,INACT command.

QUIesce
Show help on the VARY TELNET,QUIESCE command.

RESUME
Show help on the VARY TELNET,RESUME command.

STOp
Show help on the VARY TELNET,STOP command.

Examples:

Obeyfile
Show help on the VARY OBEYFILE command.

Telnet
Show the available options on the DISPLAY TELNET command.

ABENDTRAP
Show help on the VARY TELNET,ABENDTRAP command.

ACT
Show help on the VARY TELNET,ACT command.

DEBug
Show help on the VARY TELNET,DEBUG command.

INACT
Show help on the VARY TELNET,INACT command.

QUIesce
Show help on the VARY TELNET,QUIESCE command.

RESUME
Show help on the VARY TELNET,RESUME command.

STOp
Show help on the VARY TELNET,STOP command.

VARY TCPIP,*tnproc*,OBEYFILE

Purpose: Use the VARY TCPIP,*tnproc*,OBEYFILE command to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TN3270E Telnet server (Telnet) address space.

See the *z/OS Communications Server: IP Configuration Guide* 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 containing TCP/IP 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: Following are examples of 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 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 configuration statements:

```
VARY TCPIP,,O,USER99.TCPIP(OBEYFIL2)
```

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.OBEYFILE.
- The DSN= parameter cannot be a z/OS UNIX file.

VARY TCPIP,*tnproc*,TELNET

Purpose: Use the VARY TCPIP,*tnproc*,TELNET commands to control the TN3270E Telnet server (Telnet). For additional information about Telnet, see Accessing remote hosts using Telnet in the *z/OS Communications Server: IP Configuration Guide*.

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 and when to allow end users to connect. To help manage commands related to multiple ports, commands support a PORT keyword.

Tip: All parameters entered after these commands can be in any order.

The following provide details of the VARY TCPIP,*tnproc*,TELNET commands that can be used.

VARY ABENDTRAP command:

Purpose: The VARY ABENDTRAP command provides abend dumps that are based on a return code being set in a given module.

Format:

```
►►──Vary ──TCPIP──,──procname──,──Telnet──,──ABENDTRAP──,──modname──────────────►
```

```
►─┬──────────────────────┬──────────────────────────────────────────►◄
  └─rcode─┬────────────┬─┘
          └─instance───┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

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:

Purpose: 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──,──procname──,──Telnet──,──ACT──,──luname──────────────►◄
```

Parameters:

procname
: The member name of the cataloged procedure used to start the Telnet address space.

Telnet
: Directs the command to the Telnet component.

ACT
: The activate keyword.

luname
: The name of the LU you are activating. The LUNAME ALL has special meaning. It will enable all inactivated LUs.

Usage: The ACT command does not change the VTAM status of the LU. Use the INACTLUS display to show a list of LUs currently inactive.

Authorization is through the user's RACF profile containing 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:

Purpose: The VARY DEBUG command changes the DEBUG function on all active Telnet profiles.

Format:

```
►►──VARY TCPIP──,──procname──,──Telnet──,──DEBug──,──┬─────┬──────────────────────►◄
                                                     └─OFF─┘
```

Parameters:

procname
: The member name of the cataloged procedure used to start the Telnet address space.

Telnet
: Directs the command to the Telnet component.

DEBug
: The debug keyword.

OFF
: All Telnet DEBUG functions will be 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—,—procname—,—Telnet—,—INACT—,—luname————————————————►◄
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

INACT
The inactivate 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.
- The VTAM VARY NET,INACT command should be used to end the SNA LU session.
- The TCP/IP VARY DROP command should be used 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:

Purpose: The VARY QUIESCE command causes the specified port to not accept any new Telnet client connections by removing the outstanding accept for the Telnet socket. Currently established connections continue to be serviced.

Note: This command is not necessary for Obeyfile processing. An Obeyfile update will create a new profile for new connections but might not change other TCP/IP configuration values for Telnet Connections because the Telnet listening socket does not get dropped. See the *z/OS Communications Server: IP Configuration Guide* for information about how different parameter updates take effect with Obeyfile processing. For example, the TCPSENDBFRSIZE and TCPRCVBUFRSIZE parameters are unchanged for the Telnet socket unless Telnet is stopped and restarted.

A qualified port cannot be specified. For information about qualified ports, see Accessing remote hosts using Telnet in the *z/OS Communications Server: IP Configuration Guide*.

Format:

```
►►──VARY TCPIP──,──procname──,──Telnet──,──QUIesce──┬──────────────────┬──────────►◄
                                                     ├─,POrt=ALL────────┤
                                                     ├─,POrt=num────────┤
                                                     ├─,POrt=num1..num2─┤
                                                     ├─,POrt=Basic──────┤
                                                     └─,POrt=Secure─────┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

QUIesce
The QUIesce command keyword.

POrt=ALL|*num***|***num1..num2***|***num***,qual|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).
- Qualified ports can have a mixture of Basic profiles and Secure profiles. If a mixed port exists and either the Basic or Secure option is chosen, the port remains active.
- 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:

Purpose: 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 might not change other TCP/IP configuration values for Telnet Connections because the Telnet listening socket does not get dropped. See the *z/OS Communications Server: IP Configuration Guide* for information about how different parameter updates take effect with Obeyfile processing.

A qualified port cannot be specified. For information about qualified ports, see Accessing remote hosts using Telnet in the *z/OS Communications Server: IP Configuration Guide*.

Format:

```
►►──VARY TCPIP──,──procname──,──Telnet──,──RESUME──┬────────────────────┬──────────►◄
                                                   ├─,POrt=ALL──────────┤
                                                   ├─,POrt=num──────────┤
                                                   ├─,POrt=num1..num2───┤
                                                   ├─,POrt=Secure───────┤
                                                   └─,POrt=Basic────────┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

RESUME
The RESUME keyword.

POrt=ALL | *num* | *num1..num2* | *num*,**qual** | **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).
- Qualified ports can have a mixture of Basic profiles and Secure profiles. If a mixed port exists and either the Basic or Secure option is chosen, the port remains active.
- 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:

Purpose: The VARY STOP command ends the port connection and all active connections. STOP does not end all of Telnet. The command processor is still 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 information about qualified ports, see Accessing remote hosts using Telnet in the *z/OS Communications Server: IP Configuration Guide*.

Format:

```
►►──VARY TCPIP──,──procname──,──Telnet──,──STOp──┬────────────────────┬──────────►◄
                                                 ├─,POrt=ALL──────────┤
                                                 ├─,POrt=num──────────┤
                                                 ├─,POrt=num1..num2───┤
                                                 ├─,POrt=Secure───────┤
                                                 └─,POrt=Basic────────┘
```

Parameters:

procname
The member name of the cataloged procedure used to start the Telnet address space.

Telnet
Directs the command to the Telnet component.

STOp
The STOP command keyword.

POrt=ALL | *num* | *num1..num2* | *num*,qual | 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).
- Qualified ports can have a mixture of Basic profiles and Secure profiles. If a mixed port exists and either the Basic or Secure option is chosen, the port remains active.
- 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.

TSO commands

The following topics describe some of the system administrator TSO commands.

- "Using the SMSG interface" on page 227
- "MAKESITE command" on page 228
- "TESTSITE command" on page 232
- "HOMETEST command" on page 233
- "MVPXDIS command" on page 234

Using the SMSG interface

Purpose

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 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 Monitoring the status of SMTP using the SMSG command in the *z/OS Communications Server: IP User's Guide and Commands* for information 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

Purpose

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 will be 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:

   - 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.

```
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

Purpose

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.

Note: The TSO TESTSITE command uses the Pascal socket API, so VMCF must be started for the command to be successful. 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

Purpose

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.

Format

```
►►──HOMETEST──────────────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

MVPXDIS command

Purpose

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 787 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 utilizes 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 10.

Table 10. 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); only available 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 10. 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 10 on page 237).

Table 11. 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

Purpose

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 may not be the hostname 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

Purpose

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 may 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

Purpose

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

Purpose

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

Purpose

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

Purpose

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 499.
- 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 515.
- The TSO TRACERTE and z/OS UNIX **traceroute**/**otracert** commands help debug network problems. See "Traceroute" on page 521.

Netstat

The TSO NETSTAT and z/OS UNIX **netstat**/**onetstat** commands provide information about the status of the local host, including information about TCP/IP connections, network clients, gateways, and devices. TSO NETSTAT and z/OS UNIX **netstat**/**onetstat** also 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 will have to be reviewed and possibly modified when maintenance or a new release of z/OS is being installed. In every new release, the *z/OS Summary of Message and Interface Changes* updates the IP Netstat operator commands DISPLAY TCPIP,,NETSTAT, IP NETSTAT TSO commands, and IP **netstat** UNIX commands that identify the changes to the Netstat reports in that release.

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 doesn't 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 V1R9                             */
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.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.ROUTE	ROUTE / -r
EZB.NETSTAT.mvsname.tcpprocname.SLAP	SLAP / -j
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

Resource names in SERVAUTH class	Netstat options
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

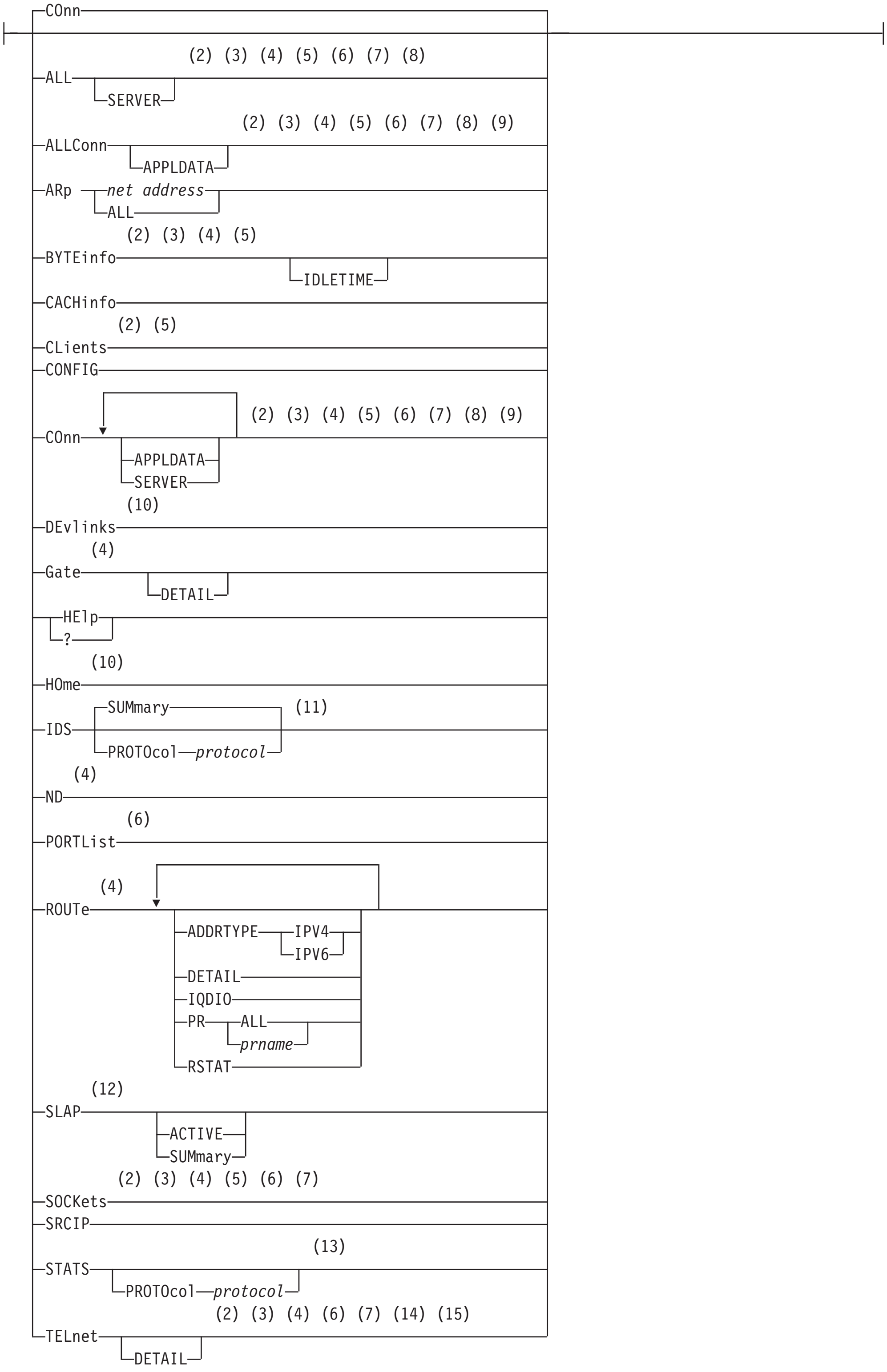
Purpose

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
ALL
SERVER
(2) (3) (4) (5) (6) (7) (8)
ALLConn
APPLDATA
(2) (3) (4) (5) (6) (7) (8) (9)
ARp
net address
ALL
BYTEinfo
IDLETIME
(2) (3) (4) (5)
CACHinfo
CLients
(2) (5)
CONFIG
COnn
APPLDATA
SERVER
(2) (3) (4) (5) (6) (7) (8) (9)
DEvlinks
(10)
Gate
DETAIL
(4)
HElp
?
HOme
(10)
IDS
SUMmary
PROTOcol
protocol
(11)
ND
(4)
PORTList
(6)
ROUTe
(4)
ADDRTYPE
IPV4
IPV6
DETAIL
IQDIO
PR
ALL
prname
RSTAT
SLAP
(12)
ACTIVE
SUMmary
SOCKets
(2) (3) (4) (5) (6) (7)
SRCIP
STATS
PROTOcol
protocol
(13)
TELnet
DETAIL
(2) (3) (4) (6) (7) (14) (15)

```
├──┬──────────────────────────────────────┬──────────────────────────────────────────────┤
   │        ┌─GRoup──────────────────────┐ │
   ├─TTLS───┼────────────────────────────┤ │
   │        ├─COnn──connid──┬────────┬───┤ │
   │        │               └─DETAIL─┘   │ │
   │        └─GRoup──┬────────┬──────────┘ │
   │                 └─DETAIL─┘            │
   ├─Up────────────────────────────────────┤
   │      (3) (4) (6) (7)                  │
   ├─VCRT──────────────────────┬────────┬──┤
   │                           └─DETAIL─┘  │
   │      (4) (6) (7)                      │
   ├─VDPT──────────────────┬────────┬──────┤
   │                       └─DETAIL─┘      │
   │          (4)                          │
   ├─VIPADCFG────────┬────────┬────────────┤
   │                 └─DETAIL─┘            │
   └─VIPADyn──┬───────────┬────────────────┘
              ├─DVIPA─────┤
              └─VIPAROUTE─┘
```

Command:

```
├──DRop ──n──────────────────────────────────────────────────────────────────────────┤
```

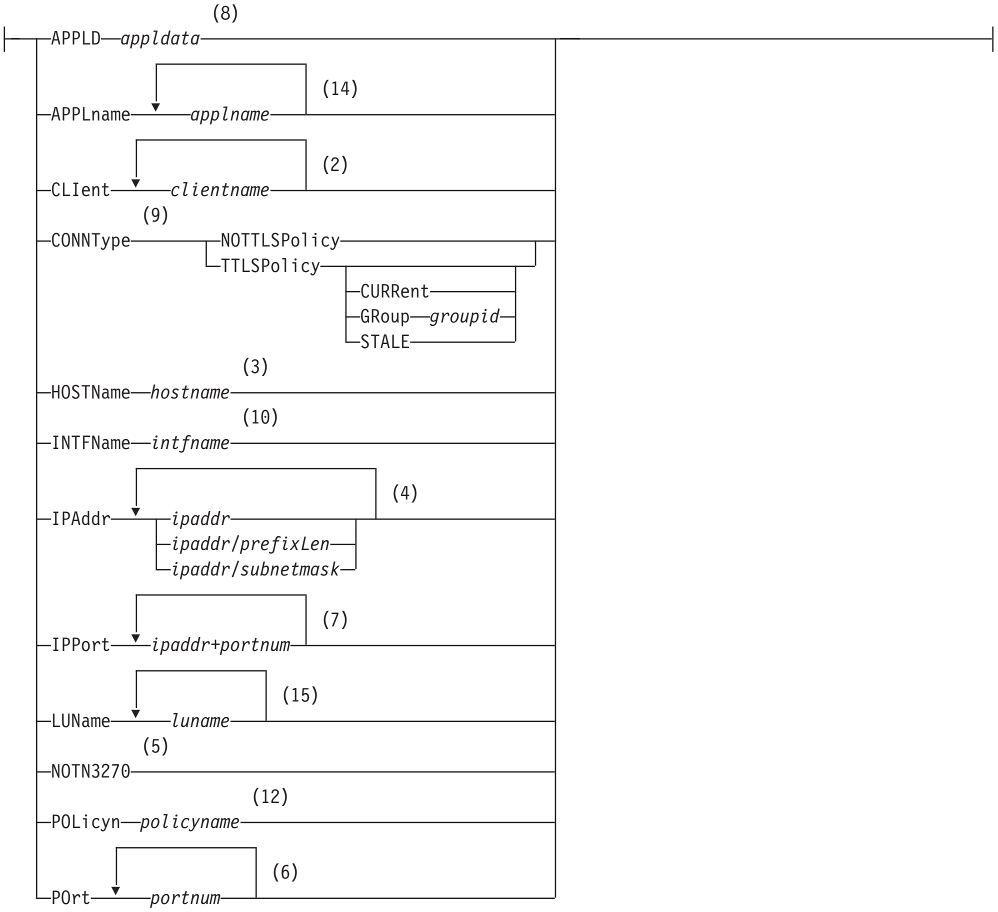
Target:

```
├──TCp tcpname──────────────────────────────────────────────────────────────────────┤
```

Output:

```
├──┬─REPort──┬─────────────────┬───────────┬──────────────────────────────────────────┤
   │         ├─DSN──dsnname────┤           │
   │         └─HLQ──hlqname────┘           │
   └─STACk──┬──────────┬─FORMat──┬─LONG──┬─┘
            └─TITLes───┘         └─SHORT─┘
```

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, SOCKets, TELnet, and VCRT.

4 The IPAddr filter is valid only with ALL, ALLConn, BYTEinfo, COnn, Gate, ND, 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 CONNType filter is valid only with ALLConn and COnn.

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

11 The valid protocol values are TCP and UDP.

12 The POLicyn filter is valid only with SLAP.

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

14 The APPLname filter is valid only with TELnet.

15 The LUName filter is valid only with TELnet.

The z/OS UNIX netstat command syntax

Purpose

Use the z/OS UNIX **netstat** command to display the network configuration and status on a local TCP/IP stack.

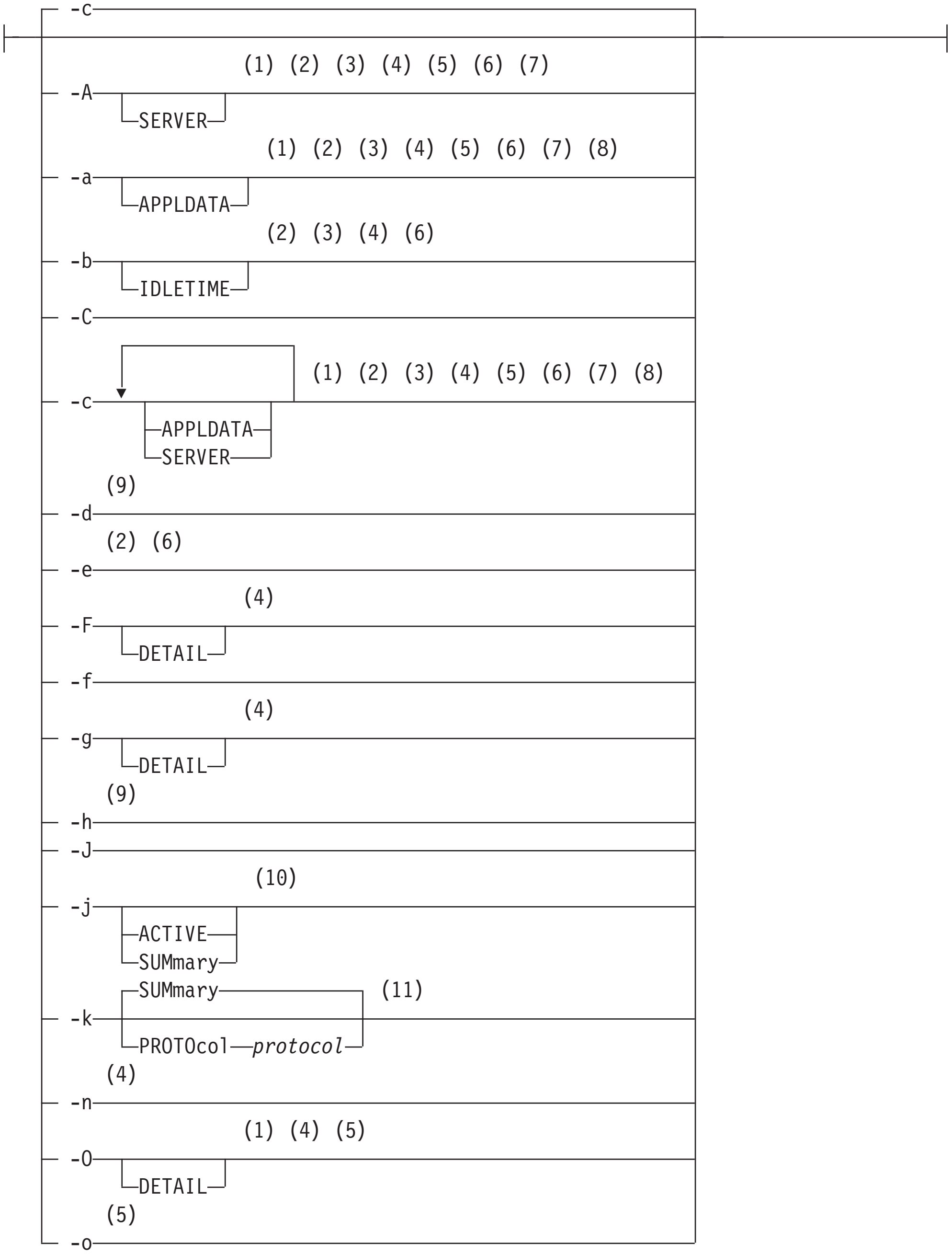
Notes:

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
-A
SERVER
(1) (2) (3) (4) (5) (6) (7)
-a
APPLDATA
(1) (2) (3) (4) (5) (6) (7) (8)
-b
IDLETIME
(2) (3) (4) (6)
-C
-c
APPLDATA
SERVER
(1) (2) (3) (4) (5) (6) (7) (8)
-d
(9)
-e
(2) (6)
-F
DETAIL
(4)
-f
-g
DETAIL
(4)
-h
(9)
-J
-j
ACTIVE
SUMmary
(10)
-k
SUMmary
PROTOcol
protocol
(11)
-n
(4)
-O
DETAIL
(1) (4) (5)
-o
(5)

```
├──┬─────────────────────────────────────────────────────┬──────────────────────────────┤
   ├─ -R─┬─net address─┬────────────────────────────────┤
   │     └─ALL─────────┘                                │
   │        ┌────────────────────────────────┐          │
   │        ▼                                │ (4)      │
   ├─ -r────┬─ADDRTYPE──┬─IPV4─┬─────────────┤          │
   │        │           └─IPV6─┘             │          │
   │        ├─DETAIL─────────────────────────┤          │
   │        ├─IQDIO──────────────────────────┤          │
   │        ├─PR──┬─ALL────┬─────────────────┤          │
   │        │     └─prname─┘                 │          │
   │        └─RSTAT──────────────────────────┘          │
   │                                   (12)             │
   ├─ -S──┬─────────────────────────────┬───────────────┤
   │      └─PROTOcol──protocol──────────┘               │
   │    (1) (2) (3) (4) (5) (6)                         │
   ├─ -s────────────────────────────────────────────────┤
   │                (1) (2) (3) (4) (5) (13) (14)       │
   ├─ -t──┬────────┬────────────────────────────────────┤
   │      └─DETAIL─┘                                    │
   ├─ -u────────────────────────────────────────────────┤
   │               (1) (3) (4) (5)                      │
   ├─ -V─┬────────┬─────────────────────────────────────┤
   │     └─DETAIL─┘                                     │
   ├─ -v─┬───────────┬──────────────────────────────────┤
   │     ├─DVIPA─────┤                                  │
   │     └─VIPAROUTE─┘                                  │
   │      ┌─GRoup─────────────────────────┐             │
   ├─ -x──┼───────────────────────────────┼─────────────┤
   │      ├─COnn──connid──┬────────┬──────┤             │
   │      │               └─DETAIL─┘      │             │
   │      └─GRoup──┬────────┬─────────────┘             │
   │               └─DETAIL─┘                           │
   └─ -?────────────────────────────────────────────────┘
```

Command:

```
├── -D n─────────────────────────────────────────────────────────────────┤
```

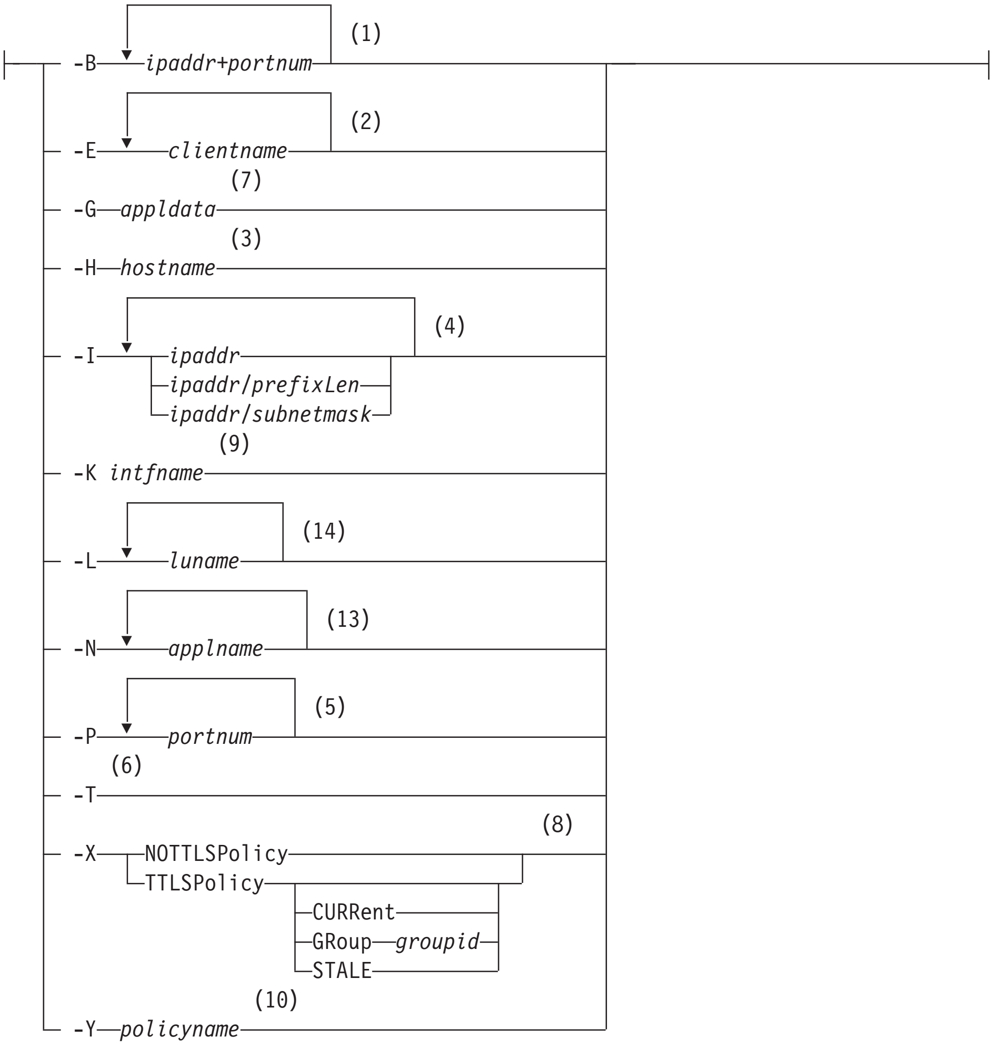
Target:

```
├── -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, -s, -t, and -V.

4 -I filter is valid only with -A, -a, -b, -c, -F, -g, -n, -O, -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 -X filter is valid only with -a, and -c.

9 -K filter is valid only with -d and -h.

10 -Y filter is valid only with -j.

11 The valid protocol values are TCP, and UDP.

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

13 -N filter is valid only with -t.

14 -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.

Report option

The following are report options that can be used with the Netstat command. If no report option is specified, then 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 274 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 298 for more details.

ARp/-R
Queries the IPv4 ARP cache information. See "Netstat ARp/-R report" on page 305 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 307 for more details.

CACHinfo/-C
Displays information about TCP connections utilizing the Cache Accelerator. See "Netstat CACHinfo/-C report" on page 313 for more details.

CLients/-e
Displays information about local users of TCP/IP services (jobnames). See "Netstat CLients/-e report" on page 315 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 317 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 339 for more details.

DEvlinks/-d
Displays the information about devices and their interfaces or links defined to the TCP/IP stack. See "Netstat DEvlinks/-d report" on page 345 for more details.

Gate/-g
Displays information about the stack routing table for IPv4 destinations. See "Netstat Gate/-g report" on page 372 for more details.

HElp or ?/-?
Displays help information for the Netstat parameters. See "Netstat HElp/-? report" on page 377 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 381 for more details.

IDS/-k
Displays information about intrusion detection services. See "Netstat IDS/-k report" on page 385 for more details.

ND/-n Displays the IPv6 Neighbor cache entries. See "Netstat ND/-n report" on page 391 for more details.

PORTList/-o
Displays the port reservation list. See "Netstat PORTList/-o report" on page 394 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 396 for more details.

SLAP/-j
Displays the QoS policy statistics. See "Netstat SLAP/-j report" on page 407 for more details.

SOCKets/-s
Displays information about each client using a socket application programming interface. See "Netstat SOCKets/-s report" on page 411 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 417 for more details.

STATS/-S
Displays TCP/IP statistics for IP, ICMP, TCP, and UDP protocols. See "Netstat STATS/-S report" on page 419 for more details.

TELnet/-t
Displays information for TN3270 Telnet server connections. See "Netstat TELnet/-t report" on page 434 for more details.

TTLS/-x
Displays Application Transparent Transport Layer Security (AT-TLS) group and connection information. See "Netstat TTLS/-x report" on page 441 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 453 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 454 for more details.

VDPT/-O

Displays the dynamic VIPA Destination Port Table information. See "Netstat VDPT/-O report" on page 462 for more details.

VIPADCFG/-F

Displays the dynamic VIPA configuration for a TCP/IP stack. See "Netstat VIPADCFG/-F report" on page 475 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 489 for more details.

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.

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*.

Output

Use the following options to specify where and in which format output should be 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.

If . . .	Then . . .
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:

```
MVS TCP/IP NETSTAT CS V1R9       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.

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 while the AT-TLS function was enabled, this includes the following:

- 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. The following are possible values for AT-TLS connection type:

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.

HOSTName/-H *hostname*
Filter the output of the ALL/-A, ALLConn/-a, BYTEinfo/-b, COnn/-c, 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 256 characters in length.

Result: 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 does not support wildcard characters.
2. Using the HOSTName filter might cause delays in the output due to resolution of the *hostname* value (depending on resolver and DNS configuration).

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 up to 16 characters in length.

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.

The INTFName filter can also be used to format a specific OSAENTA trace interface by specifying EZANTA*portname* value, where the *portname* value is the name that was specified on the PORTNAME keyword in the TRLE statement for the OSA that is being traced.

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*. 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 ALL/-A, ALLConn/-a, BYTEinfo/-b, COnn/-c, Gate/-g, ND/-n, ROUTe/-r, SOCKets/-s, TELnet/-t, VCRT/-V, VDPT/-O, and VIPADCFG/-F reports 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 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.

Notes:

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:

1. For ALL/-A, ALLConn/-a, COnn/-c, and TELnet/-t options, *ipaddr* can be either the local or remote IP address. For 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 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.
2. 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 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.

Restrictions:

1. The IPAddr/-I filter for VCRT/-V, VDPT/-O, and VIPADCFG/-F options does not support wildcard characters.

2. The IPAddr/-I filter for an IPv6 address does not support wildcard characters.
3. For an IPv4-only stack, only IPv4 *ipaddr* values are accepted.
4. For the ND/-n option, an IPv4 *ipaddr* value is not accepted.

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 – 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:

- 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.
- 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.

Guideline: The port number can be either a local or remote port. For the SOCKets option, the port can be a port to which the socket is bound or connected.

Except for POrt/-P, INTFName/-K, CONNType/-X TTLSPolicy GRoup *groupid*, HOSTname/-H, 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, care should be taken 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, the character string should be surrounded by 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

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 FACILITY 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 370 for detailed information.

Netstat report details and examples

The following general concepts apply:

General concepts

In order to fully understand the following concepts and fields, you need to have some general knowledge of TCP/IP. See the IBM Redbook ***TCP/IP tutorial and Technical Overview***, GG24-3376 for more information.

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.

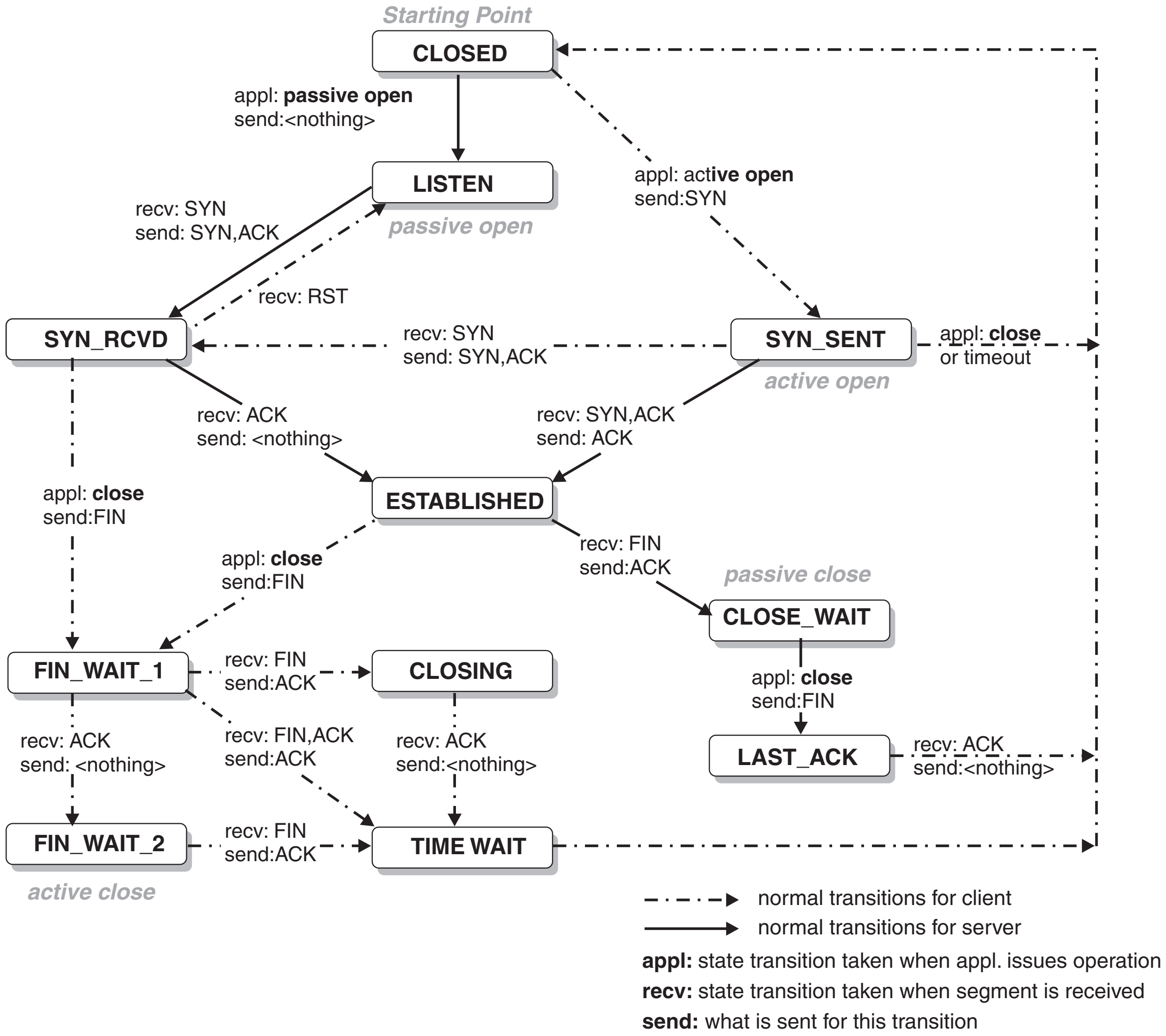


Figure 1. TCP state transition diagram

Table 12. 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 will find a local TCP server's listening socket.
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.

Table 12. TCP state transition description table (continued)

TCP connection state	Abbreviation in MVS console	Abbreviation in TSO or UNIX shell	Description
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 should normally be 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 should normally be 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.

Table 12. TCP state transition description table (continued)

TCP connection state	Abbreviation in MVS console	Abbreviation in TSO or UNIX shell	Description
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.
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.

A server program's listening socket will have 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 will have 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:

- 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:

- 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.

Time stamp

The time stamp displayed in the header for each Netstat report is local time. The time field displayed in reports ALL/-A, BYTEinfo/-b, CLients/-e, SLAP/-j, UP/-u, and VIPADyn/-v is Coordinated Universal Time (UTC).

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 "Output" on page 263 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 "Output" on page 263 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:

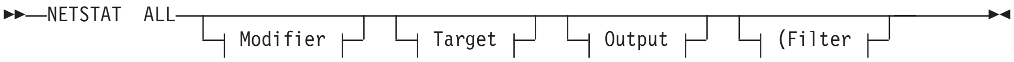
```
netstat -b > byteinfo
```

The file byteinfo is created in your current directory containing the screen output shown previously.

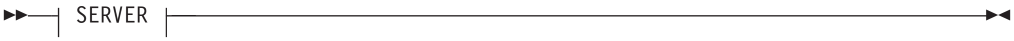
Netstat ALL/-A report

Purpose: 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:



Modifier:

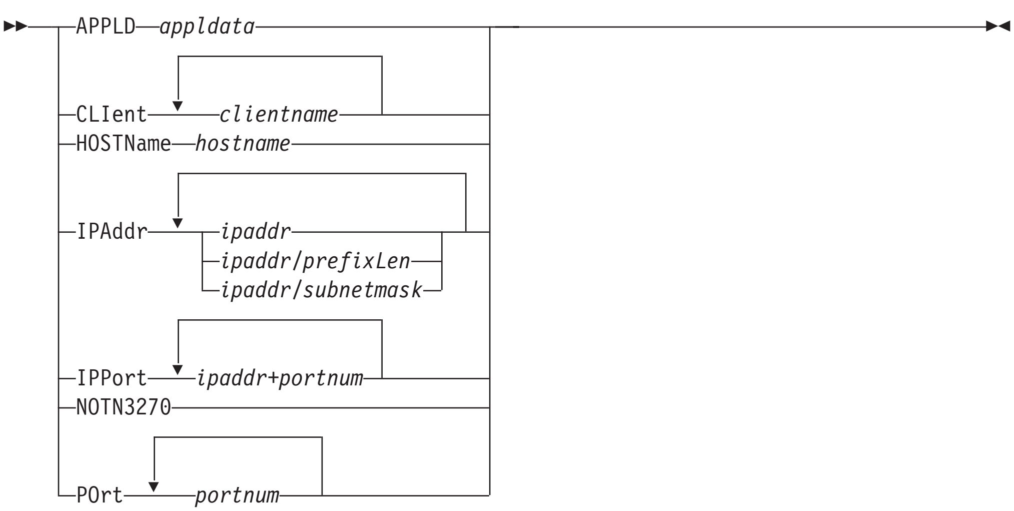


SERVER
Provide detailed information only for TCP connections in the listen state.

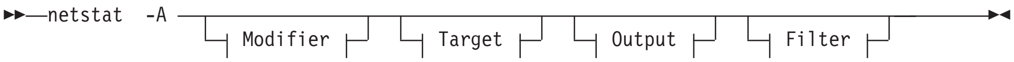
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

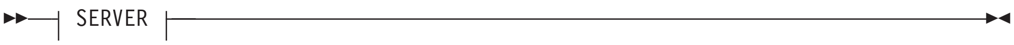
Filter:



z/OS UNIX syntax:



Modifier:

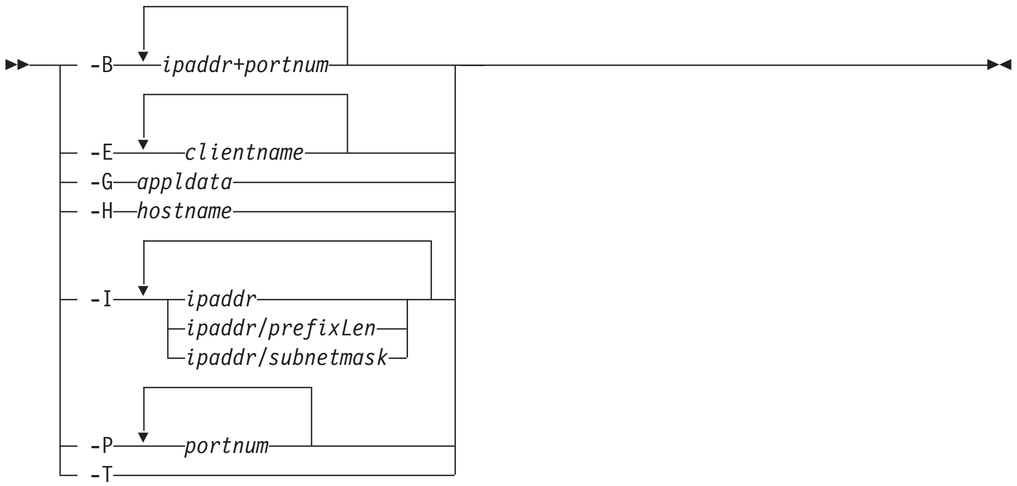


SERVER
Provide detailed information only for TCP connections in the listen state.

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 256 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.

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 – 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.

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.

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, care should be taken 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, the character string should be surrounded by 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 IPPORT=127.0.0.1+21 TCP TCPCS
   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 V1R9       TCPIP NAME: TCPCS            17:40:36
Client Name: FTPD1                    Client Id: 0000003B
Local Socket: 0.0.0.0..21             Foreign Socket: 0.0.0.0..0
  Last Touched:       17:09:22          State:              Listen
  BytesIn:            0000000000        BytesOut:           0000000000
  SegmentsIn:         0000000000        SegmentsOut:        0000000000
  RcvNxt:             0000000000        SndNxt:             0000000000
  ClientRcvNxt:       0000000000        ClientSndNxt:       0000000000
  InitRcvSeqNum:      0000000000        InitSndSeqNum:      0000000000
  CongestionWindow:   0000000000        SlowStartThreshold: 0000000000
  IncomingWindowNum:  0000032768        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:             65536
  SockOpt:            80                TcpTimer:           00
  TcpSig:             00                TcpSel:             20
  TcpDet:             40                TcpPol:             00
  QOSPolicyRuleName:
  TTLSPolicy:         No
  RoutingPolicy:      No
  ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
  ConnectionsIn:      0000000000        ConnectionsDropped: 0000000000
  CurrentBacklog:     0000000000        MaximumBacklog:     0000000010
  CurrentConnections: 0000000300        SEF:                098
  Quiesced:           Dest
  SharePort: WLM
    RawWeight:        63                NormalizedWeight:   15
    Abnorm:           10                Health:             100
    RawCP:  060       RawzAAP:  000     RawzIIP:  040
    PropCP: 040       PropzAAP: 000     PropzAAP: 023
----
```

```
Client Name: TCPCS                    Client Id: 0000000C
Local Socket: 9.67.115.5..23          Foreign Socket: 9.27.11.182..4665
  Last Touched:       16:46:15          State:              Establsh
  BytesIn:            0000001062        BytesOut:           0000000480
  SegmentsIn:         0000000019        SegmentsOut:        0000000019
  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
  QOSPolicyRuleName:
  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
  Application Data:   EZBTNSRV TCPM1001 TSO10002 ET ST14S
----
Client Name: SYSLOGD1                 Client Id: 00000010
Local Socket: 0.0.0.0..514            Foreign Socket: *..*
  Last Touched:       16:46:29
  BytesIn:            0000000000        BytesOut:           0000000000
  DgramIn:            0000000000        DgramOut:           0000000000
  MaxSendLim:         0000065535        MaxRecvLim:         0000065535
  SockOpt:            00                DSField:            00
  QOSPolicyRuleName:
  RoutingPolicy:      No
----
Client Name: APPV4                    Client Id: 00000015
Local Socket: 0.0.0.0..2049           Foreign Socket: 9.42.103.99..1234
  Last Touched:       16:00:29
  BytesIn:            0000000200        BytesOut:           0000000100
  DgramIn:            0000000010        DgramOut:           0000000005
  MaxSendLim:         0000065535        MaxRecvLim:         0000065535
  SockOpt:            00                DSField:            00
  QOSPolicyRuleName:
  RoutingPolicy:      Yes
    RoutingTableName: prTab4
    RoutingRuleName:  SecLow4

----
```

IPv6 enabled or request for LONG format:

```
NETSTAT ALL
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:32:05
Client Name: FTPD1                    Client Id: 0000004A
  Local Socket: ::..21
  Foreign Socket: ::..0
    BytesIn:            00000000000000000000
    BytesOut:           00000000000000000000
    SegmentsIn:         00000000000000000000
    SegmentsOut:        00000000000000000000
    Last Touched:       14:27:36          State:              Listen
    RcvNxt:             0000000000        SndNxt:             0000000000
    ClientRcvNxt:       0000000000        ClientSndNxt:       0000000000
    InitRcvSeqNum:      0000000000        InitSndSeqNum:      0000000000
    CongestionWindow:   0000000000        SlowStartThreshold: 0000000000
    IncomingWindowNum:  0000032768        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:             65536
    SockOpt:            0000              TcpTimer:           00
    TcpSig:             00                TcpSel:             20
    TcpDet:             C0                TcpPol:             00
    QOSPolicyRuleName:
    TTLSPolicy:         Yes
      TTLSRule:         TTLSRule2
      TTLSGrpAction:    TTLSGrpAction1
    RoutingPolicy:      No
    ReceiveBufferSize:  0000016384        SendBufferSize:     0000016384
    ConnectionsIn:      0000000000        ConnectionsDropped: 0000000000
    CurrentBacklog:     0000000000        MaximumBacklog:     0000000004
    CurrentConnections: 0000000300        SEF:                098
    Quiesced:           Dest
    SharePort: WLM
      RawWeight:        63                NormalizedWeight:   15
      Abnorm:           10                Health:             100
----
```

```
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
    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
    QOSPolicyRuleName:
    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
    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
    Last Touched:       16:00:29
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicyRuleName:
    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
    Last Touched:       16:00:29
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicyRuleName:
    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
    Last Touched:       14:27:42
    MaxSendLim:         0000065535        MaxRecvLim:         0000065535
    SockOpt:            00000000          DSField:            00
    QOSPolicyRuleName:
    RoutingPolicy:      No
    ReceiveDataQueued:  0000345655        ReceiveMsgCnt:      0000004564
      OldQDate:         09/15/06          OldQTime:           03:36:32
    TTLSPolicy:         No
    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 “General concepts” on page 269 for a detailed description.

Client ID
See the Client ID or Connection Number information in “General concepts” on page 269 for a detailed description.

Local Socket
See the Local Socket information in “General concepts” on page 269 for a detailed description.

Foreign Socket
See the Foreign Socket information in “General concepts” on page 269 for a detailed description.

Last touched
See the Last touched time information in “General concepts” on page 269 for a detailed description.

State
Describes the state of the TCP connection. See “TCP connection status” on page 269 for more information.

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.

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.

SegmentsIn
The number of segments received for this connection. A segment is the group of data bytes contained in a TCP packet.

SegmentsOut
The number of segments sent for this connection.

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.

InitRcvSegNum
The first sequence number that was received from the remote stackhost when establishing the connection.

InitSndSegNum

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 will be 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 acknowledgement 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, the value will be one of the following:

- 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 will be 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.

| **SndWnd**

| 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 cause the TCP/IP stack to periodically send empty packets to the remote stack to make sure the connection is still alive.

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.

QOSPolicyRuleName
The name of the QOS policy rule 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 will be no rule or actions listed.
- **TTLSPolicy: Yes** indicates one of the following:
  - 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.

ReceiveDataQueued

The number of bytes of data on the receive queue from the remote application yet to be read. 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.

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. 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.

The SendDataQueued information is not displayed for a connection that is in LISTEN state.

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 Configuration 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.

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 socket option SO_CLUSTERCONNTYPE was requested. For more information about the cluster connection type, see the *z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference*.

08 .... 1...
If the SO_CLUSTERCONNTYPE socket option 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 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 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 was issued for this socket, this bit indicates that the connection partners are not in the same cluster.

Any other value

Is used for diagnostic purposes only under the direction of IBM Service personnel.

ConnectionsIn

The number of connections that a server has accepted. 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.

CurrentBacklog

The number of connection requests currently in the backlog queue. These are connection requests that have been received by the server when the backlog queue was not full, but have not yet finished connection establishment.

MaximumBacklog

The maximum number of connection requests a server will maintain waiting to finish the connection establishment. Connection requests that are received when the maximum number of requests is already on the backlog queue are generally discarded. The higher the maximum backlog queue, the more simultaneous connection requests a server can handle without having to drop requests.

CurrentConnections

The number of currently established connections to the server.

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.

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 or not 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. 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 will change to No.

SharePort

Indicates that multiple TCP listening servers are sharing the same port. 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 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 should be 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 the distribution method is not SERVERWLM, then the value of this field is 0. For more information on 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 regarding 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, then the value of this field is 100. If the distribution method is not SERVERWLM, then the value of this field is 100. For more information on 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.

ProcCP
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).

ProczAAP
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).

ProczIIP
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).

- The following fields are displayed for a UDP socket entry:

Client Name
See the Client name or User ID information in "General concepts" on page 269 for a detailed description.

Client ID
See the Client ID or Connection Number information in "General concepts" on page 269 for a detailed description.

Local Socket
See the Local Socket information in "General concepts" on page 269 for a detailed description.

Foreign Socket
See the Foreign Socket information in "General concepts" on page 269 for a detailed description.

Last touched time
See the Last touched time information in "General concepts" on page 269 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.

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, the value will be one of the following:
  - 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 will be 0.

QOSPolicyRuleName
The name of the Policy rule 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 will be `Exclude`.

Netstat ALLConn/-a report

Purpose: 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 “Target” on page 263 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 251 or “Output” on page 263.

Filter:

```
►►──┬─APPLD──appldata──────────────────────────┬──────────────────►◄
    │                                          │
    │        ┌────────────────┐                │
    │        ▼                │                │
    ├─CLIent───clientname─────┴────────────────┤
    ├─HOSTName──hostname───────────────────────┤
    │                                          │
    │        ┌──────────────────────┐          │
    │        ▼                      │          │
    ├─IPAddr───┬─ipaddr─────────────┼──────────┤
    │          ├─ipaddr/prefixLen───┤          │
    │          └─ipaddr/subnetmask──┘          │
    │                                          │
    │        ┌────────────────┐                │
    │        ▼                │                │
    ├─IPPort───ipaddr+portnum─┴────────────────┤
    ├─NOTN3270─────────────────────────────────┤
    │                                          │
    │      ┌──────────────┐                    │
    │      ▼              │                    │
    ├─POrt───────portnum──┴────────────────────┤
    └─CONNType──┬─NOTTLSPolicy─────────────────┘
                └─TTLSPolicy──┬─────────────┬─┘
                              ├─CURRent─────┤
                              ├─GRoup──groupid─┤
                              └─STALE───────┘
```

z/OS UNIX syntax:

```
►►──netstat  -a───────────────────────────────────────────────────►

►──┬──────────────┬──┬────────────┬──┬────────────┬──┬────────────┬──►◄
   └─┤ Modifier ├─┘  └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ Filter ├─┘
```

Modifier:

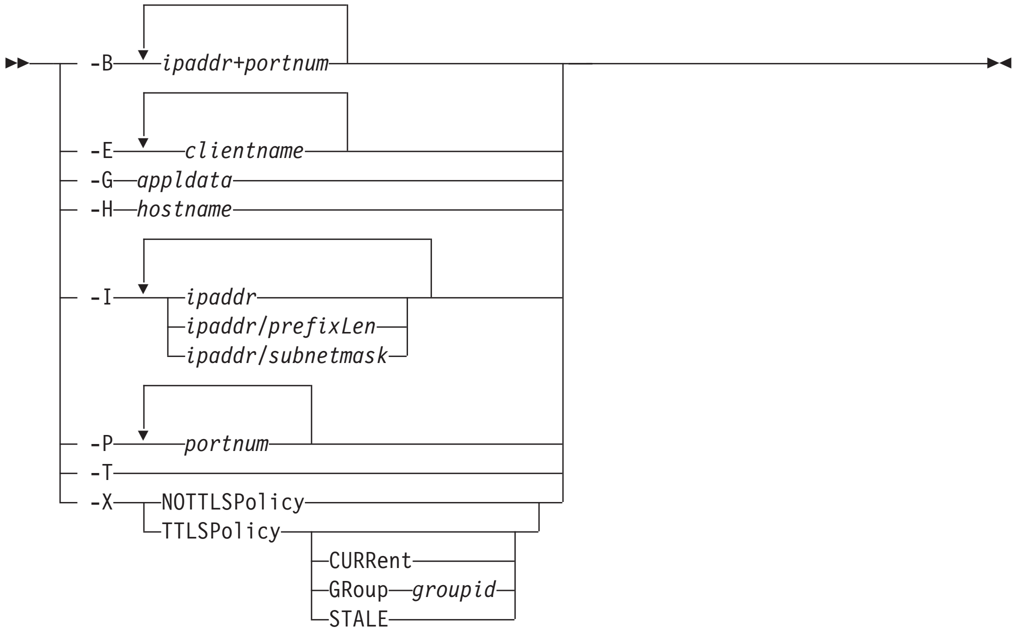
```
►►──APPLDATA──────────────────────────────────────────────────────►◄
```

APPLDATA
Provides application data in the output report.

Target: Provide the report for a specified TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 256 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.

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 – 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.

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.

| 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, care should be taken 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, the character string should be surrounded by 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’**.

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:

- 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. The following are possible values for AT-TLS connection type:

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.

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 V1R9       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 V1R9       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 "General concepts" on page 269 for a detailed description.

Conn See the Client ID or Connection Number information in "General concepts" on page 269 for a detailed description.

Local Socket
: See the Local Socket information in “General concepts” on page 269 for a detailed description.

Foreign Socket
: See the Foreign Socket information in “General concepts” on page 269 for a detailed description.

State
: See the TCP connection status and UDP socket status information in “General concepts” on page 269 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 in the *z/OS Communications Server: IP Configuration 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

Purpose: 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 link.

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 “Target” on page 263 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 251 or “Output” on page 263.

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 “Target” on page 263 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 256 or “Output” on page 263.

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 V1R9       TCPIP NAME: TCPCS           12:48:54
Querying ARP cache for address 201.2.10.32
Link: SZ_TR1            IBMTR: 08005A0D97A2
Route info: 0270

Querying ARP cache for address 201.2.10.31
Link: SZ_TR1            IBMTR: 08005A0D97A2
Route info: 0270

Querying ARP cache for address 9.67.128.1
Link: IQDIOLNKC0010203  IPAQIDIO

Querying ARP cache for address 9.67.1.8
Link: OSA90LINK1        NSAP: 39999999999999999999ABCDEFABCD1234567890
```

```
NETSTAT ARP 201.2.10.32
MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS           12:48:54
Querying ARP cache for address 201.2.10.32
Link: 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.

Link The link name.

Link Type
The link type. This field is not displayed for HiperSockets links.

MAC address
The MAC address associated with the IP address. This field is not displayed for HiperSockets links.

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

Purpose: 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 "Target" on page 263 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 251 or "Output" on page 263.

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 “Target” on page 263 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 256 or “Output” on page 263.

Filter:

```
             ┌──────────────┐
             ▼              │
►►──┬─ -E──────clientname───┴──┬───────────────────────────────────►◄
    ├─ -H──hostname────────────┤
    │        ┌────────────────────────┐
    │        ▼                        │
    ├─ -I──────┬─ipaddr───────────────┼─┤
    │          ├─ipaddr/subnetmask────┤ │
    │          └─ipaddr/prefixLen─────┘ │
    └─ -T───────────────────────────────┘
```

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 256 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.

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, care should be taken 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, the character string should be surrounded by 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 V1R9       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 V1R9       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 V1R9       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 V1R9       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 = *nnnnnnnnnnnnnnnnnn* x 1000)

Report field descriptions:

User Id
: See the Client name or User ID information in “General concepts” on page 269 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.

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.

LPort
: See the Local port description in “General concepts” on page 269 for a detailed description.

Foreign Socket
See the Foreign socket information in “General concepts” on page 269 for a detailed description.

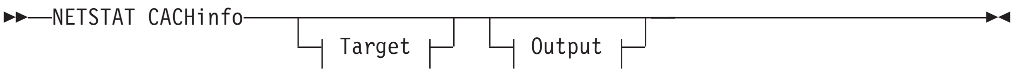
State See the TCP connection status and UDP socket status information in “General concepts” on page 269 for a detailed description.

IdleTime
The time interval between the current time and the last time the connection was touched. See Last touched time in “General concepts” on page 269 for a detailed description of the last touched time.

Netstat CACHinfo/-C report

Purpose: Displays statistics for TCP listening sockets that are utilizing the Fast Response Cache Accelerator (FRCA). For more information about the FRCA, see the *z/OS Communications Server: IP Configuration Guide*.

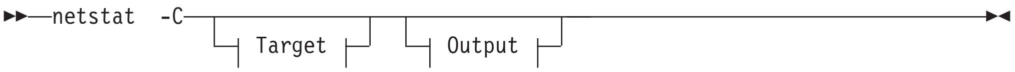
TSO syntax:



Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 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 251 or “Output” on page 263.

z/OS UNIX syntax:



Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 V1R9       TCPIP Name: TCPCS           22:27:39
Client: USER1147        Listening socket: 0.0.0.0..80
  MaxCacheSize:       0000000100  CurrCacheSize:      0000000001
  MaxNumObjects:      0000000010  CurrNumObjects:     0000000001
  NumConns:           0000000002  ConnsProcessed:     0000000001
  ConnsDeferred:      0000000001  ConnsTimedOut:      0000000000
  RequestsProcessed:  0000000001  IncompleteRequests: 0000000000
  NumCacheHits:       0000000002  NumCacheMisses:     0000000000
  NumUnprodCacheHits: 0000000000
```

IPv6 enabled or request for LONG format:

```
NETSTAT CACHINFO
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           22:27:39
Client: USER1147
  Listening socket: 0.0.0.0..80
  MaxCacheSize:       0000000100  CurrCacheSize:      0000000001
  MaxNumObjects:      0000000010  CurrNumObjects:     0000000001
  NumConns:           0000000002  ConnsProcessed:     0000000001
  ConnsDeferred:      0000000001  ConnsTimedOut:      0000000000
  RequestsProcessed:  0000000001  IncompleteRequests: 0000000000
  NumCacheHits:       0000000002  NumCacheMisses:     0000000000
  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.

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 connection requests that were at least partially processed by the Cache Accelerator.

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

Purpose: 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 “Target” on page 263 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 251 or “Output” on page 263.

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 "Target" on page 263 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 256 or "Output" on page 263.

Filter:

```
          ┌────────────────┐
          ▼                │
►►──┬─ -E────clientname────┴─┬─────────────────────────────────────────►◄
    └─ -T────────────────────┘
```

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, care should be taken 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, the character string should be surrounded by 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 V1R9       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 "General concepts" on page 269 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 are 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 "General concepts" on page 269 for a detailed description.

Netstat CONFIG/-f report

Purpose: 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 "Target" on page 263 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 251 or "Output" on page 263.

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 "Target" on page 263 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 256 or "Output" on page 263.

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 V1R9       TCPIP Name: TCPCS           12:55:20
TCP Configuration Table:
DefaultRcvBufSize:  00016384  DefaultSndBufSize: 00016384
DefltMaxRcvBufSize: 00262144
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:      600
TTLS:               Yes

UDP Configuration Table:
DefaultRcvBufSize: 00065535  DefaultSndBufSize: 00065535
CheckSum:          Yes
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: No
MultiPath:  Conn   PathMtuDsc: No     DevRtryDur:  0000000090
DynamicXCF: Yes
  IpAddr/PrefixLen: 193.9.200.3/28    Metric: 01
  SecClass: 8
IQDIORoute: Yes       QDIOPriority: 1
TcpStackSrcVipa: 201.1.10.10

SMF Parameters:
Type 118:
  TcpInit:      01   TcpTerm:    02   FTPClient:    03
  TN3270Client: 00   TcpIpStats: 05
Type 119:
  TcpInit:      No   TcpTerm:    No   FTPClient:    Yes
  TcpIpStats:   Yes  IfStats:    Yes  PortStats:    Yes
  Stack:        Yes  UdpTerm:    Yes  TN3270Client: Yes

Global Configuration Information:
TcpIpStats: Yes  ECSALimit: 0002047K  PoolLimit: 2096128K
MlsChkTerm: No   XCFGRPID:  11   IQDVLANID: 27
SegOffLoad: Yes  SysplexWLMPoll: 060
ExplicitBindPortRange:  05000-06023
Sysplex Monitor:
  TimerSecs:  60    Recovery:  Yes    DelayJoin:  Yes  AutoRejoin: Yes
  MonIntf:    Yes   DynRoute:  Yes
zIIP:
  IPSecurity: Yes

Network Monitor Configuration Information:
PktTrcSrv: Yes  TcpCnnSrv: Yes   MinLifTim: 3   SmfSrv: Yes

Data Trace Setting:
JobName: *           TrRecCnt: 00000009  Length: FULL
IpAddr:  *                 SubNet: *

Autolog Configuration Information: Wait Time: 5
ProcName: FTPD       JobName: FTPD1     DelayStart: Yes
  ParmString:
```

IPv6 enabled or request for LONG format:

```
NETSTAT CONFIG
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           12:55:20
TCP Configuration Table:
DefaultRcvBufSize:  00016384  DefaultSndBufSize: 00016384
DefltMaxRcvBufSize: 00262144
MaxReTransmitTime:  120.000   MinReTransmitTime: 0.500
RoundTripGain:      0.125     VarianceGain:      0.250
VarianceMultiplier: 2.000     MaxSegLifeTime:    30.000
DefaultKeepALive:   00000120  DelayAck:          Yes
RestrictLowPort:    No        SendGarbage:       No
TcpTimeStamp:       Yes       FinWait2Time:      600
TTLS:               Yes


UDP Configuration Table:
DefaultRcvBufSize: 00065535  DefaultSndBufSize: 00065535
CheckSum:          Yes
RestrictLowPort:   No        UdpQueueLimit:     Yes

IP Configuration Table:
Forwarding: Yes    TimeToLive: 00064  RsmTimeOut:  00060
ArpTimeout: 01200  MaxRsmSize: 65535  Format:      Long
IgRedirect: Yes    SysplxRout: No     DoubleNop:   No
StopClawEr: No     SourceVipa: No
MultiPath:  No     PathMtuDsc: No     DevRtryDur:  0000000090
DynamicXCF: No
IQDIORoute: No
TcpStackSrcVipa: No

IPv6 Configuration Table:
Forwarding:    Yes  HopLimit:   00064  IgRedirect:  Yes
SourceVipa:    No   MultiPath:  No     IcmperrLim:  00003
IgRtrHopLimit: No
IpSecurity: Yes
DynamicXCF: Yes
  IpAddr: 2001:0db8::9:67:115:5
  IntfId: 0009:0067:0011:0001
  SrcVipaInt: ipv6srcvipa
  SecClass: 8
TcpStackSrcVipa: ipv6stksrcvipa

SMF Parameters:
Type 118:
  TcpInit:      01   TcpTerm:    02   FTPClient:    03
  TN3270Client: 00   TcpIpStats: 05
Type 119:
  TcpInit:      No   TcpTerm:    No   FTPClient:    Yes
  TcpIpStats:   Yes  IfStats:    Yes  PortStats:    Yes
  Stack:        Yes  UdpTerm:    Yes  TN3270Client: Yes

Global Configuration Information:
TcpIpStats: No  ECSALimit: 0000000K  PoolLimit: 0000000K
MlsChkTerm: Yes  XCFGRPID:  14  IQDVLANID: 117
SegOffLoad: Yes  SysplexWLMPoll: 060
ExplicitBindPortRange:  05000-06023
Sysplex Monitor:
  TimerSecs:  60    Recovery:  Yes    DelayJoin:  Yes  AutoRejoin: Yes
  MonIntf:    Yes   DynRoute:  Yes
zIIP:
  IPSecurity: Yes

Network Monitor Configuration Information:
PktTrcSrv: Yes  TcpCnnSrv: Yes   MinLifTim: 3   SmfSrv: Yes

Data Trace Setting:
JobName: *          TrRecCnt: 00000009  Length: FULL
IpAddr/PrefixLen: 9::44/128

Autolog Configuration Information: Wait Time: 5
ProcName: FTPD      JobName: FTPD1     DelayStart: Yes
  ParmString:
```

Report field descriptions:

- **TCP Configuration Table**

  Display the following configured TCP information that is defined in the TCPCONFIG profile statement. For more information about each TCP parameter, see the TCPCONFIG profile statement information in the *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 16 384 (16 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 16384 (16 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 256 KB; the default is 16384 (16 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 16384 (16 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 is 512 KB, and the default 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.

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 – 35 791 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 acknowledgements 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 acknowledgements 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.

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).

Note: The values displayed in the MaxReTransmitTime, MinReTransmitTime, RoundTripGain, VarianceGain, VarianceMultiplier, and MaxSegLifeTime fields are the actual default values that were assigned by the TCP/IP stack and cannot be configured externally using the TCPCONFIG profile statement. The values can be overridden on a per-destination basis using either the BEGINROUTES configuration statement, the old GATEWAY configuration statement, or OMPROUTE's configuration file.

- **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).

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 utilize 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 utilize 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 – 86 400; 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 – 65 535; the default value is 65 535.

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 or that OMPROUTE has been started with an IPv4 routing protocol configured).

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 4 294 967 295. 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.

IQDIORoute
Indicates whether HiperSockets Accelerator is enabled for this TCP/IP stack. This field can have the following values:

Yes Indicates that IQDIOROUTING is in effect (the IQDIOROUTING parameter was specified in the IPCONFIG profile statement).

No Indicates that IQDIOROUTING is not in effect (the NOIQDIOROUTING parameter was specified in the IPCONFIG profile statement or is in effect by default).

QDIOPriority
Indicates which QDIO outbound priority level should be used if the HiperSockets Accelerator is routing 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.

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

- **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 utilize 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 utilize 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 or OMPROUTE has been started with an IPv6 routing protocol configured).

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).

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

- **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 records are created when TCP connections are established. 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).

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 or not 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 1 048 576 bytes). If the K suffix is used, then the value displayed must be in the range 10 240K – 2 096 128K inclusive, or 0K. If the M suffix is used, the value displayed must be in the range 10M to 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 1 048 576 bytes). If the K suffix is used, then the value displayed must be in the range 10 240K to 2 096 128K 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.

SegOffload

Indicates whether TCP segmentation offload is enabled or disabled. This field can have the following values:

Yes Indicates that TCP segmentation will be offloaded to OSA-Express interfaces that support segmentation offload (the SEGMENTATIONOFFLOAD parameter was specified on the GLOBALCONFIG profile statement).

No Indicates that segmentation will be performed by the TCP/IP stack (the NOSEGMENTATIONOFFLOAD parameter was specified on the GLOBALCONFIG profile statement or the value was set by default).

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.

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.

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 regarding the problem, leaves the sysplex group, and inactivates 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 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 are 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.

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).

IPSecurity
Indicates whether the stack is configured to displace CPU cycles for IPSec workload onto a zIIP. The value `No` indicates that IPSec CPU cycles are not being displaced to a zIIP. The value `Yes` indicates that IPSec CPU cycles are displaced to a zIIP as long as at least one zIIP device is online. The zIIP online or offline status can be displayed using the MVS D M=CPU command. See displaying system configuration information details in *z/OS MVS System Commands* for more information about displaying processor status.

- **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 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.

SmfSrv
Indicates whether the real-time SMF information service is enabled or disabled. The value `Yes` indicates that SMFSERVICE is in effect (the SMFSERVICE parameter was specified in the NETMONITOR profile statement). The value `No` indicates that SMFSERVICE is not in effect (the NOSMFSERVICE parameter was specified in the NETMONITOR profile statement or is in effect by default).

- **Autolog Configuration Information**

WaitTime
The time specified on the AUTOLOG statement that represents the length of time TCP/IP should wait 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 should start.

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.

DelayStart
Indicates whether TCP/IP should delay starting this procedure until after the TCP/IP stack has joined the sysplex group. This field can have the following values:

Yes Indicates that the TCP/IP stack does not start 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).

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).

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.

- **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.

Netstat COnn/-c report

Purpose: 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───┘
```

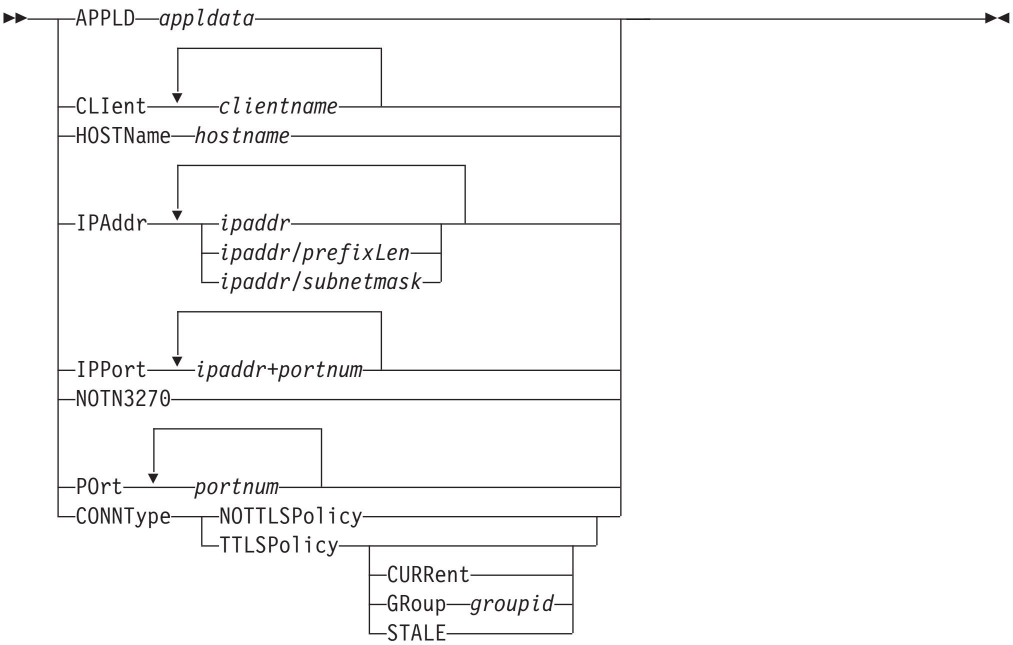
APPLDATA
: Provides application data in the output report.

SERVER
: Provide detailed information only for TCP connections in the listen state.

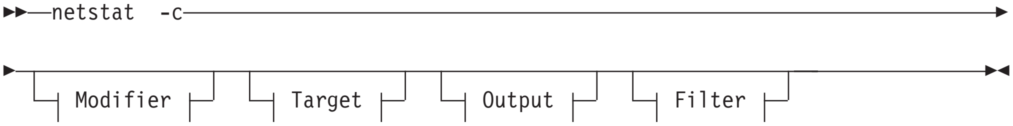
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

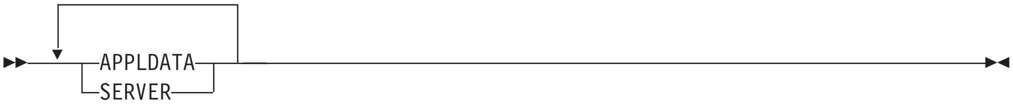
Filter:



z/OS UNIX syntax:



Modifier:



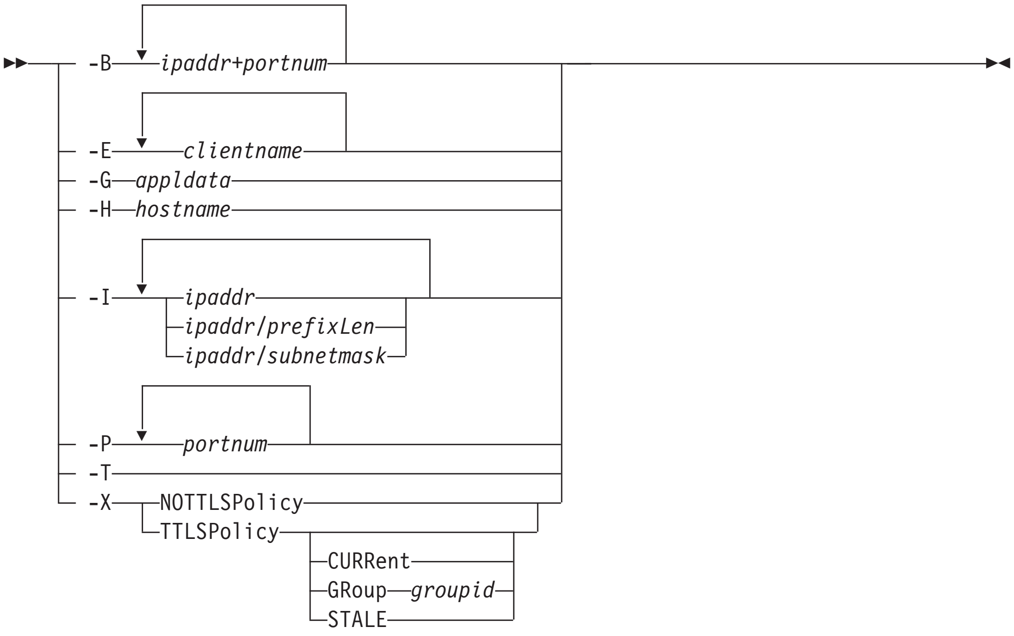
APPLDATA
Provides application data in the output report.

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 “Target” on page 263 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 256 or “Output” on page 263.

Filter:



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 256 characters in length.

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 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 *ipaddr*
ipaddr/prefixlength
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.

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 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.

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 – 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.

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.

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 the following:

- 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 a 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. The following are possible values for AT-TLS connection type:

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, care should be taken 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, the character string should be surrounded by 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 V1R9      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 V1R9       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 "General concepts" on page 269 for a detailed description.

Conn See the Client ID or Connection Number information in "General concepts" on page 269 for a detailed description.

Local Socket
See the Local Socket information in "General concepts" on page 269 for a detailed description.

Foreign Socket
See the Foreign Socket information in "General concepts" on page 269 for a detailed description.

State See the TCP connection status and UDP socket status information in "General concepts" on page 269 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 Configuration 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 DEvlinks/-d report

Purpose: Displays information about devices and defined interfaces or links defined to the TCP/IP stack.

TSO syntax:

```
►►──NETSTAT DEvlinks──┬────────────┬──┬────────────┬──┬─────────────┬──────────►◄
                      └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ (Filter ├─┘
```

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "Target" on page 263 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 251 or "Output" on page 263.

Filter:

```
►►──INTFName──intfname───────────────────────────────────────────────►◄
```

z/OS UNIX syntax:

```
►►──netstat  -d──┬────────────┬──┬────────────┬──┬────────────┬──────────►◄
                 └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ Filter ├─┘
```

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See "Target" on page 263 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 256 or "Output" on page 263.

Filter:

```
►►── -K──intfname──────────────────────────────────────────────────►◄
```

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 filter can also be used to format a specific OSAENTA trace interface by specifying EZANTA*portname,* where the *portname* value is the name specified on the PORTNAME keyword in the TRLE statement for the OSA that is being traced.

If . . .	Then . . .	Else . . .
A network resource has been coded in TCPIP.PROFILE using the DEVICE/LINK/HOME statements	Use the *intfname* value specified on the LINK profile statement.	Use the *intfname* value specified on the INTERFACE profile statement.

Restriction: The INTFName/-K filter value does not support wildcard characters.

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 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 -p tcpcs6 -K OSAQDIOLINKI
```

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 V1R9       TCPIP Name: TCPCS           14:23:39
DevName: LOOPBACK          DevType: LOOPBACK
  DevStatus: Ready
  LnkName: LOOPBACK          LnkType: LOOPBACK   LnkStatus: Ready
    NetNum: n/a  QueSize: n/a
    ActMtu: 65535
  BSD Routing Parameters:
    MTU Size: 00000             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
  BSD 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
    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
    NetNum: n/a  QueSize: n/a  Speed: 0000000100
    VMacAddr:   000629DC21BC  VMacOrigin: Cfg  VMacRouter: All
    IpBroadcastCapability: No
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    ActMtu: 1492
    VLANid: 1260                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)      InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 8                      MonSysplex: Yes
  BSD Routing Parameters:
    MTU Size: 00000             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
    NetNum: n/a  QueSize: 0
    ActMtu: Unknown
    SecClass: 8                      MonSysplex: Yes
  BSD Routing Parameters:
    MTU Size: 00000             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
    NetNum: n/a  QueSize: n/a
    ActMtu: 2600
    SecClass: 8                     MonSysplex: No
  BSD Routing Parameters:
    MTU Size: 00000            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
IPv4 LAN Group Summary
 LanGroup: 001

   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   --------   ---------
   Link1     Active      Link1      yes
   Link2     Not Active  Link1      no

 LanGroup: 002

   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   --------   ---------
   Link3     Active      Link3      no
   Link4     Active      Link4      yes

 LanGroup: 003
```

```
   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   --------   ---------
   Link5     Active      Link5      yes
   Link6     Active      Link6      no
```

IPv6 enabled or request for LONG format:

```
NETSTAT DEVLINKS
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:23:39
DevName: LOOPBACK          DevType: LOOPBACK
  DevStatus: Ready
  LnkName: LOOPBACK          LnkType: LOOPBACK   LnkStatus: Ready
    NetNum: n/a  QueSize: n/a
    ActMtu: 65535
  BSD Routing Parameters:
    MTU Size: 00000             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
    NetNum: n/a  QueSize: n/a
    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
  BSD 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
    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
    NetNum: n/a  QueSize: n/a  Speed: 0000000100
    VMacAddr:   000629DC21BC  VMacOrigin: Cfg  VMacRouter: All
    IpBroadcastCapability: No
    CfgRouter: Non                   ActRouter: Non
    ArpOffload: Yes                  ArpOffloadInfo: Yes
    ActMtu: 1492
    VLANid: 1260                     VLANpriority: Enabled
    DynVLANRegCfg: Yes               DynVLANRegCap: No
    ReadStorage: GLOBAL (8064K)      InbPerf: Balanced
    ChecksumOffload: Yes             SegmentationOffload: Yes
    SecClass: 8                      MonSysplex: Yes
  BSD Routing Parameters:
    MTU Size: 00000             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
    NetNum: n/a  QueSize: n/a  Speed: 0000000100
    VMacAddr:   000629DC21BC  VMacOrigin: Cfg  VMacRouter: All
    MacAddress: 000629DC21BC
    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
    SecClass: 8                      MonSysplex: Yes
  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

DevName: IUTSAMEH          DevType: MPCPTP
  DevStatus: Not Active
  IntfName: V6SAMEH          IntfType: MPCPTP6   IntfStatus: Not Active
    NetNum: n/a  QueSize: n/a
    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

DevName: VIPAV6            DevType: VIPA
  DevStatus: Ready
  IntfName: VIPAV6           IntfType: VIPA6     IntfStatus: Ready
    NetNum: n/a  QueSize: n/a
  Packet Trace Setting:
    Protocol: *                 TrRecCnt: 00000000  PckLength: FULL
    SrcPort:  *                 DestPort: *         PortNum: *
    IpAddr:   *                 SubNet:   *
  Multicast Specific:
    Multicast Capability: No
IPv4 LAN Group Summary

 LanGroup: 001
   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   -------    ---------
   Link1     Active      Link1      yes
   Link2     Not Active  Link1      no

 LanGroup: 002
   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   --------   ---------
   Link3     Active      Link3      no
   Link4     Active      Link4      yes

 LanGroup: 003
   LnkName   LnkStatus   ArpOwner   VipaOwner
   -------   ---------   --------   ---------
   Link5     Active      Link5      yes
   Link6     Active      Link6      no


IPv6 LAN Group Summary
 LanGroup: 004

   IntfName   IntfStatus   NDOwner    VipaOwner
   --------   ----------   --------   ---------
   Intf1      Active       Intf1      yes
   Intf2      Not Active   Intf1      no

 LanGroup: 0052
   IntfName   IntfStatus   NDOwner    VipaOwner
   --------   ----------   --------   ---------
   Intf3      Active       Intf3      no
   Intf4      Active       Intf4      yes

 LanGroup: 0063
   IntfName   IntfStatus   NDOwner    VipaOwner
   --------   ----------   --------   ---------
   Intf5      Active       Intf5      yes
   Intf6      Active       Intf6      no
```

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

This field is the device name configured on the DEVICE statement or generated based on the INTERFACE statement.

DevType

This field is the device type configured on the DEVICE statement or generated based on the INTERFACE statement.

DevNum

This field is the device number configured on the DEVICE statement. This field is significant only for device types CTC, CLAW, LCS, and CDLC.

DevStatus

The field is the device status. The following list describes the 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.
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 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 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 only for IPAQENET links and for IPAQENET6 interfaces. This field is displayed only if virtual MAC (VMAC) has been configured.

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 and for IPAQENET6 interfaces for which virtual MAC (VMAC) has been configured. The following are 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 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 on Virtual MAC router attributes. The following are possible values:

All Corresponds to the ROUTEALL keyword. Indicates that all IP traffic destined to the Virtual MAC will be forwarded by the OSA-Express 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 by this TCP/IP stack will be forwarded by the OSA-Express.

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 configured on the LINK statement or the interface name configured on the INTERFACE statement.

LnkType/IntfType
This field is the link type configured on the LINK statement or the interface type configured on the INTERFACE statement.

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.

Link/Interface status	Description
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 will be 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 will have produced 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.

NetNum
This field indicates the adapter number specified on the LINK statement. This field is significant only for CTC and LCS links. A value of n/a is displayed for these links or interfaces other than CTC and LCS.

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. A value of n/a is displayed for these links or interfaces other than ATM, LCS, and IPAQENET6.

Speed This field indicates the interface speed (in million bits per second) reported by the device. This field is significant only for ATM, IPAQENET, and IPAQTR links and IPAQENET6 interfaces, and only if the link or interface is active.

MAC address order (MacAddrOrder)
This field indicates the canonical option (CANON/NONCANON) specified on the LINK statement. This field is significant only for token-ring links.

SrBridgingCapability
This field indicates whether the link supports source route bridging. This field is significant only for token-ring links.

IpBroadcastCapability
This field indicates whether the link is broadcast capable. This field is significant only for links on LCS and MPCIPA devices.

ArpBroadcastType
This field indicates the ARP broadcast option (ALLRINGSBCAST/ LOCALBCAST) specified on the LINK statement. This field is significant only for token-ring links.

ArpOffload
This field indicates whether ARP processing is being offloaded to the adapter. This field is significant only for active links that support ARP offload.

ArpOffloadInfo
This field indicates whether the adapter reports ARP offload data to TCP/IP. If so, then the ARP cache data can be displayed with Netstat ARP/-R even though the ARP function is being offloaded. This field is significant only for active links that support ARP offload.

BSD Routing Parameters

This field is significant only for IPv4 links. Use the BSDROUTINGPARMS statement to define the characteristics of each link including the subnet mask.

MTU Size

The MTU size represents the MTU configured in BSDROUTINGPARMS, or, if using OMPROUTE, on an OSPF_INTERFACE, a RIP_Interface, or an INTERFACE statement. This value is the largest packet size that can be sent over this link. All routes using this link must have an MTU value no larger than this value.

Metric The interface routing metric represents the configured cost_metric for usage by dynamic routing daemons for routing preferences. If using NCPROUTE, it is configured using the cost_metric value in the BSDROUTINGPARMS profile statement. If using OMPROUTE, it is configured using the in_metric parameter in the RIP_INTERFACE statement or using the cost0 parameter in the OSPF_INTERFACE statement. If using DYNAMICXCF, it is configured using the cost_metric value in the IPCONFIG DYNAMICXCF statement. For more information, see the corresponding parameter descriptions in the *z/OS Communications Server: IP Configuration Reference*.

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.

SubnetMask

The subnet mask associated with 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 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 of *, 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.

IpAddr

The IP address from the IPADDR keyword of the PKTTRACE command or * if not specified.

SubNet

The IP subnet mask from the SUBNET keyword of the PKTTRACE command or * 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 will be 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 will have produced 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. The

value of this field is always Yes for point-to-point interfaces. For LCS and MPCIPA links and IPAQENET6 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.

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.

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 will use 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 will be 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 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 will be placed in the Hop Count field of the IP header for outgoing IP packets. This value was obtained from a received Router Advertisement. This field is significant only for IPAQENET6 interfaces.

If . . .	Then . . .	Else . . .
The IGNOREROUTERHoplimit is specified from the IPCONFIG6 profile statement	IgRtrHopLimit field will be indicated with `Yes` and the global HopLimit value (displayed on the Netstat CONFIG report) will be used instead of this value.	The global HopLimit value (displayed on the Netstat CONFIG report) will be used.

CfgMtu
The MTU value configured on the INTERFACE statement (or `None` if no MTU was configured). This field is significant only for IPAQENET6 interfaces.

ActMtu
The largest MTU supported by an active link or interface. If the link or

interface is inactive, then this field displays as ’Unknown’. This field is significant for all links and interfaces except virtual ones.

VLANid

This field is significant only for active IPAQENET and IPAQIDIO links or IPAQENET6 and IPAQIDIO6 interfaces that support Virtual LAN IDs.

For OSA-Express QDIO, a VLAN ID of 0000 indicates that the VLANID parameter was not specified on the LINK or INTERFACE profile statement for the link or interface, or that the OSA-Express does not support VLAN IDs.

For an IPAQIDIO link or IPAQIDIO6 interface, a VLAN ID of 0000 indicates one of the following conditions:

- The VLANID parameter was not specified on the LINK or INTERFACE profile statement for the link or interface.
- If the link or interface was dynamically generated as part of dynamic XCF HiperSockets processing, the IQDVLANID parameter was not specified on the GLOBALCONFIG statement.
- HiperSockets does not support VLAN IDs.

If an OSA-Express 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. If this field is zero, the IP packets are being null tagged, with a null VLANID but with priority information. If an OSA-Express is not active or does not support either VLAN priority or Virtual LAN IDs, then this field is not displayed.

VLANpriority

This field is significant only for active IPAQENET links or IPAQENET6 interfaces that support Virtual LAN priorities. If an OSA-Express is active and supports Virtual LAN priorities, this field indicates that all IP packets through this OSA-Express link or interface from this stack are being tagged with a VLAN priority. If an OSA-Express is not active or does not support either VLAN priority or Virtual LAN IDs, then this field is not displayed.

Enabled

Indicates that all IP packets through this OSA-Express link or interface are being tagged with a VLAN priority. See the *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 supports VLAN priority, but currently no VLAN priority values are defined. If the VLAN ID is zero, all IP packets through this OSA-Express link or interface are not VLAN tagged. If VLAN ID is nonzero, all IP packets are VLAN tagged, but with only VLAN IDs and not VLAN priority.

DynVLANRegCfg

This field is significant only for IPAQENET and IPAQENET6 interfaces. It 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 either selected by default or was specified on the LINK or INTERFACE statement.

DynVLANRegCap
This field is significant only for IPAQENET and IPAQENET6 interfaces. It indicates whether the OSA feature represented by the LINK or INTERFACE statement is capable of supporting dynamic VLAN ID registration. The possible values are:

Yes Indicates that the OSA feature is capable of supporting dynamic VLAN ID registration.

No Indicates that the OSA feature is not capable of supporting dynamic VLAN ID registration.

Unknown
Indicates that the dynamic VLAN ID registration capability of the OSA feature is unknown because the LINK or INTERFACE is not yet active.

ChecksumOffload
This field is significant only for active IPAQENET links. This field indicates whether checksum offload support is in effect and is displayed only when the adapter is enabled for checksum offload. The possible value is:

Yes Indicates that the adapter is enabled for checksum offload for IPv4 packets.

SegmentationOffload
This field is significant only for active IPAQENET links. This field indicates whether TCP segmentation offload support is in effect and is displayed only when the adapter is enabled for segmentation offload. The possible value is:

Yes Indicates the adapter is enabled for TCP segmentation offload for IPv4 packets.

SecClass
This field is significant for all IPv4 links and IPv6 interfaces except virtual and loopback. It indicates the security class value that was defined using the SECCLASS parameter on the LINK or INTERFACE profile statement. Valid security class values are displayed as a value in the range 1 – 255.

MonSysplex
Indicates whether the status of this link or interface is being monitored by Sysplex Autonomics. This field is significant for all IPv4 links 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.

ReadStorage
This field is significant only for active IPAQENET, IPAQTR, and IPAQIDIO links and for IPAQIDIO6 and IPAQENET6 interfaces. This field indicates the amount of storage (in kilobytes) being used for read processing.

InbPerf
This field is significant only for active IPAQENET links, IPAQTR links, and IPAQENET6 interfaces. This field indicates how frequently the adapter should interrupt the host. The possible values are:

MinCPU
Indicates that the adapter is using a static interrupt-timing value that minimizes host interrupts, and therefore minimizes host CPU consumption.

MinLatency
Indicates that the adapter is using a static interrupt-timing value that minimizes latency delay by more aggressively presenting received packets to the host.

Balanced
Indicates that the adapter is using a static interrupt-timing value that strikes a balance between MinCPU and MinLatency.

Dynamic
Indicates that the stack and the adapter are dynamically updating the frequency with which the adapter should interrupt the host for inbound traffic.

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 sum of the unicast, multicast, and broadcast inbound packets received over an interface.

Inbound Packets In Error
Number of inbound packets discarded due to an error validating the packet.

Inbound Packets Discarded
Number of inbound packets discarded due to an out-of-storage condition.

Inbound Packets With No Protocol
Number of inbound packets discarded due to an unknown protocol type.

BytesOut
Number of bytes transmitted over an interface.

Outbound Packets
The sum of the unicast, multicast, and broadcast outbound packets transmitted over an interface.

Outbound Packets In Error
Number of outbound packets discarded due to errors other than an out-of-storage condition.

Outbound Packets Discarded
Number of outbound packets discarded due to an out-of-storage condition.

IPv4 LAN Group Summary
The IPv4 LAN group summary lists links that are takeover candidates for each other. The stack creates a LAN group when it detects redundant connectivity to a LAN. For each link in the LAN group, this summary displays which link owns ARP responsibility for that link. The summary also displays which link 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.

LnkName/IntfName
The link name configured on the LINK statement or the interface name configured on the INTERFACE statement.

LnkStatus/IntfStatus
The link or interface status. Valid values are Active or Not Active.

ArpOwner
The link name that owns ARP responsibility for this link in the LAN group. An active link 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.

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 are 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 are 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, 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.

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.

Netstat DRop/-D command

Purpose: 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 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 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 wish to terminate the server itself, but wish 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 wish to terminate all socket activity from a specific sockets application, the application should be terminated using the appropriate mechanism 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.

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 V1R9       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 V1R9       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

Purpose: 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────────────────────────────────────────────────────────────────►◄
```

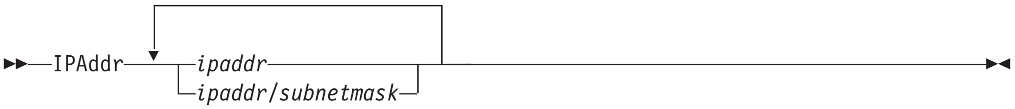
DETAIL

Displays the general IPv4 routing information, the metric or cost of use for the route, and the MVS specific configured parameters for each route.

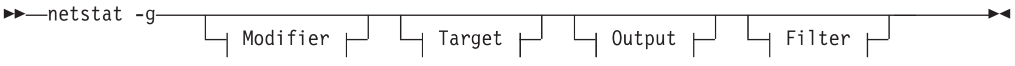
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "Target" on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user's terminal. For other options, see "The TSO NETSTAT command syntax" on page 251 or "Output" on page 263.

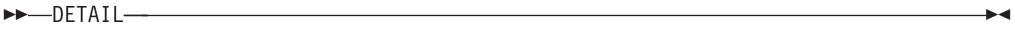
Filter:



z/OS UNIX syntax:



Modifier:

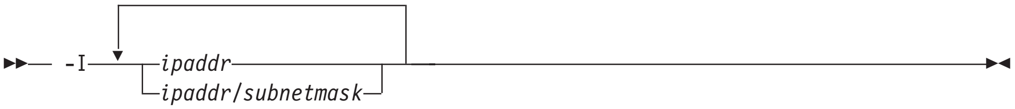


DETAIL
: Displays the general IPv4 routing information, plus the metric or cost of use for the route, and the MVS specific configured parameters for each route.

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

Filter:



Filter description:

IPAddr/-I *ipaddr*
ipaddr/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, care should be taken 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, the character string should be surrounded by 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″**.

Notes:

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 onetstat CS V1R9       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 onetstat CS V1R9       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 since 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 since 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 since 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 since 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 will be 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

Purpose: 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)
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
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
HOSTNAME  - Filter the output of ALL, ALLCONN, BYTEINFO, CONN,
            SOCKETS, TELNET and VCRT reports using the specified
            host name
INTFNAME  - Filter the output  DEVLINKS and HOME reports using the
            specified link or interface name
IPAddr    - Filter the output of ALL, ALLCONN, BYTEINFO, CONN, GATE,
            ND, 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
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
-n  - Display the IPv6 Neighbor cache entries
-O  - Display the dynamic VIPA Destination Port Table
-o  - Display port reservation list
-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, -e, -c, -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, -s, -t, and -V reports
      using the specified host name
-I    Filter the output of -A, -a, -b, -c, -F, -g, -n, -r, -s, -t, -O,
      and -V reports using the specified IP address
-K    Filter the output of -d, and -h reports using the specified
      link or interface 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
-T    Filter the output of -A, -a, -b, -c, -e, and -s reports
      excluding TN3270 server connections
-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

Purpose: Displays information about each home IP address and its associated link or interface name.

TSO syntax:

```
►►──NETSTAT HOme──┬────────────┬──┬────────────┬──┬────────────┬──────────────►◄
                  └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ (Filter ├─┘
```

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "Target" on page 263 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 251 or "Output" on page 263.

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 "Target" on page 263 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 256 or "Output" on page 263.

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.

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 V1R9       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
```

IPv6 enabled or request for LONG format:

```
NETSTAT HOME
MVS TCP/IP NETSTAT CS V1R9       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:   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::9:67:115:5
    Type:   Global
    Flags:
  Address:  fe80::6:2900:1dc:217c
    Type:   Link_Local
    Flags:  Autoconfigured
  Address:  fec0::6:2900:6dc:217c
    Type:   Site_Local
    Flags:  Autoconfigured
  Address:  50c9:c2d4::6:2900:6dc:217c
    Type:   Global
    Flags:  Autoconfigured
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
```

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:

P Primary interface.

I An internally generated dynamic VIPA that is not advertised to routing daemons. This will be displayed for dynamic VIPAs created on target stacks for sysplex distributor or on stacks that are the endpoint for connections where the dynamic VIPA has moved to another stack.

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
An internally generated dynamic VIPA that is not advertised to routing daemons. This will be displayed for dynamic VIPAs created on target stacks for sysplex distributor or on stacks that are the endpoint for connections where the dynamic VIPA has moved to another stack.

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, Link_Local, or Site_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.

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, Link_Local, or Site_Local.

Reason
Reason the IP address is unavailable:

Duplicate address detection in progress
Duplicate address detection to determine if another node is currently using the IP address that is in progress. The IP address will be 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
: A prefix address is defined and the interface ID will be appended to the prefix to create the full IP address. The interface ID is not available until the interface is successfully started.

For more information about the home list, see the *z/OS Communications Server: IP Configuration Reference*.

Netstat IDS/-k report

Purpose: 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 “Target” on page 263 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 251 or “Output” on page 263.

z/OS UNIX syntax:

```
►►──netstat -k──┬──────────────┬──┬────────────┬──┬────────────┬───────────────►◄
                └─┤ 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 -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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.

Note: The format of the Netstat IDS/-k report is not affected by IPv6 enablement nor by the Output option.

```
NETSTAT IDS
MVS TCP/IP NETSTAT CS V1R9       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
Traffic Regulation:
  TCP
    ConnRejected: 3           PlcActive: Y
  UDP
    PckDiscarded: 0           PlcActive: Y
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
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 V1R9       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
Traffic Regulation:
  TCP
    ConnRejected: 3           PlcActive: Y
  UDP
    PckDiscarded: 0           PlcActive: Y
Active Interface Floods
  IntfName: ETH1
    DiscardCnt: 1828        DiscardRate: 57   Duration: 68
```

```
NETSTAT IDS PROTOCOL TCP
MVS TCP/IP NETSTAT CS V1R9       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
```

```
NETSTAT IDS PROTOCOL UDP
MVS TCP/IP NETSTAT CS V1R9       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
```

Report field descriptions:

SUMmary

Display summary information about intrusion detection services. The following describes the information displayed by the SUMmary option.

- **For Scan Detection:**

  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.

  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:**

  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 or exceptstats is not specified in the policy, the value of this field is 0.

  Interval
  The current statistics interval or 0 if statistics or exceptstats is not specified in the policy.

- **For Traffic Regulation:**

  ConnRejected
  The number of TCP connections rejected by Traffic Regulation since the TCPIP stack was started.

PckDiscarded
The number of UDP packets discarded by Traffic Regulation since the TCPIP 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 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 link or interface name 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 Scan event 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 ibm-idsTRtcpLimitScope:PORT_INSTANCE was specified and this data applies only to this application.

N Indicates that ibm-idsTRtcpLimitScope:PORT_INSTANCE was not specified. The MxApp and MxHst information applies to all applications using this port that do not have a separate rule with PORT_INSTANCE.

TrCorr The traffic regulation constrained state correlator. A value of 0 indicates the application is not constrained.

MxHst
The total number of connections rejected since the last policy change due to a source IP exceeding the ibm-idsTRtcpPercentage of available connections it is allowed.

MxApp
The total number of connections rejected since the last policy change because the ibm-idsTRtcpTotalConnections limit was exceeded.

SynFlood

Y Indicates a syn flood is in progress.

UdpDestSocket
The destination IP address and port.

Discarded
The total number of packets since last policy change discarded because the ibm-idsTRudpQueueSize was exceeded.

Netstat ND/-n report

Purpose: 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.

TSO syntax:

```
►►──NETSTAT ND──┬──────────┬──┬──────────┬──┬──────────┬──────────────────►◄
                └┤ Target ├┘  └┤ Output ├┘  └┤ (Filter ├┘
```

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 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 251 or “Output” on page 263.

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 "Target" on page 263 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 256 or "Output" on page 263.

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 will be treated as an IPv6 address. If an IPv4-mapped IPv6 address is entered as an IPAddr/-I value, there will be 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 V1R9       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
```

Report field descriptions:

Neighbor's IP address

IntfName
Interface name where the neighbor cache entry exists.

IntfType
Interface type.

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

Purpose: Displays the port reservation list. For ports reserved by the PORTRANGE profile statement, only one output line is displayed for each range.

TSO syntax:

```
►►──NETSTAT PORTList──┬──────────┬──┬──────────┬──┬───────────┬──────────►◄
                      └┤ Target ├┘  └┤ Output ├┘  └┤ (Filter ├┘
```

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "Target" on page 263 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 251 or "Output" on page 263.

Filter:

```
           ┌─────────┐
►►──POrt───▼─portnum─┴─────────────────────────────────────────────────►◄
```

z/OS UNIX syntax:

```
►►──netstat -o──┬──────────┬──┬──────────┬──┬──────────┬────────────────►◄
                └┤ Target ├┘  └┤ Output ├┘  └┤ Filter ├┘
```

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See "Target" on page 263 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 256 or "Output" on page 263.

Filter:

```
          ┌─────────┐
►►── -P───▼─portnum─┴──────────────────────────────────────────────────►◄
```

Filter description:

POrt/-P *portnum*
: Filter the output of the PORTList/-O report using the specified port number *portnum*. You can enter up to six filter values.

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 V1R9       TCPIP Name: TCPCS           15:24:23
Port# Prot User     Flags    Range       IP Address
----- ---- ----     -----    -----       ----------
00020 TCP  FTPD1    D                    9.67.113.10
00021 TCP  FTPD1    DA
00023 TCP  TCPCS    DA
00025 TCP  SMTP     DA
04000 TCP  OMVS     DABU
00514 UDP  SYSLOGD1 DA
04020 UDP  OMVS     DAB                  9.67.43.70
04030 UDP  *        DA
05000 UDP  MUD      DAR      05000-05002
```

IPv6 enabled or request for LONG format:

```
NETSTAT PORTLIST
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           15:24:23
Port# Prot User     Flags    Range
----- ---- ----     -----    -----
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
04002 TCP  OMVS     DABU
      BindSpecific: ::6:2900:1dc:21bc
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, edfgdgdfgdgddgdgdgfdfddgdffg the PORT or PORTRANGE profile statements in the *z/OS Communications Server: IP Configuration Reference*.

Port# For ports reserved by the PORT profile statement, this value is the number of the port that was reserved. For ports 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 – 65 535.

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 “General concepts” on page 269 for detailed descriptions.

Flags The flags represent parameter values defined on the PORT or PORTRANGE profile statement.

A Autolog

B Bind

D DelayAcks

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.

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.

Netstat ROUTe/-r report

Purpose: 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─┘      │
    └─RSTAT────────────────┘
```

ADDRTYPE IPV4 | IPV6

Display the specified IP type routing information.

IPV4 Display IPv4 routing information.

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.

IQDIO

Displays the HiperSockets Accelerator routing table. This parameter is mutually exclusive with the PR and RSTAT parameters.

PR Displays policy-based routing tables. This parameter is mutually exclusive with the IQDIO parameter.

ALL Displays all policy-based routing tables.

prname

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

Restrictions:

- The PR modifier does not support IPv6 routes. If the PR modifier is used with an ADDRTYPE IPV6 value, no information is displayed.
- 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 606.

RSTAT

Displays all of the 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. The MTU value that is displayed in this report is the configured value that was defined using the MTU parameter in the ROUTE statement, or the default value for the specified interface type. This parameter is mutually exclusive with the IQDIO parameter.

When used without the PR modifier, all static routes that are defined as replaceable in the main routing table are displayed. When used with the PR modifier and keyword ALL, all static routes that are defined as replaceable in all policy-based routing tables are displayed. When a policy-based route table name is specified with the PR modifier, all static routes that are defined as replaceable in the specified policy-based routing table are displayed.

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 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 251 or “Output” on page 263.

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─┘     │
    └─RSTAT──────────────┘
```

ADDRTYPE IPV4 | IPV6
Display the specified IP type routing information.

IPV4 Display IPv4 routing information.

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.

IQDIO
Displays the HiperSockets Accelerator routing table. This parameter is mutually exclusive with the PR and RSTAT parameters.

PR Displays policy-based routing tables. This parameter is mutually exclusive with the IQDIO parameter.

ALL Displays all policy-based routing tables.

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

Restrictions:

- The PR modifier does not support IPv6 routes. If the PR modifier is used with an ADDRTYPE IPV6 value, no information is displayed.
- 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 606.

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. The MTU value that is displayed in this report is the configured value that was defined using the MTU parameter in the ROUTE statement or the default value for the specified interface type. This parameter is mutually exclusive with the IQDIO parameter.

When used without the PR modifier, all static routes that are defined as replaceable in the main routing table are displayed. When used with the PR modifier and keyword ALL, all static routes that are defined as replaceable in all policy-based routing tables are displayed. When a policy-based route table name is specified with the PR modifier, all static routes that are defined as replaceable in the specified policy-based routing table are displayed.

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See "Target" on page 263 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 256 or "Output" on page 263.

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, care should be taken 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, the character string should be surrounded by 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 will be treated as an IPv6 address. If an IPv4-mapped IPv6 address is entered as an IPAddr/-I value, there will be 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 V1R9        TCPIP Name: TCPCS           14:24:09
Destination       Gateway          Flags     Refcnt  Interface
-----------       -------          -----     ------  ---------
Default           9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32    0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32    0.0.0.0          UH        000000  OSAQDIOLINK
127.0.0.1/32      0.0.0.0          UH        000002  LOOPBACK
```

```
NETSTAT ROUTE DETAIL

MVS TCP/IP NETSTAT CS V1R9        TCPIP Name: TCPCS           14:03:13
Destination       Gateway          Flags     Refcnt  Interface
-----------       -------          -----     ------  ---------
Default           9.67.115.1       UGS       000000  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 V1R9        TCPIP NAME: TCPCS            17:40:36
IPv4 Destinations
Destination       Gateway          Interface
-----------       -------          ---------
9.67.1.9/32       0.0.0.0          OSA00LINK1
```

```
NETSTAT ROUTE IQDIO

MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS            09:51:02
Destination      Gateway          Interface
-----------      -------          ---------
9.67.1.9/32      0.0.0.0          LIQDIO1
```

```
NETSTAT ROUTE PR prtable1

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: Yes  MultiPath: Conn(Policy)
  DynamicXCFRoutes:    No
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UH        000000  OSAQDIOLINK
```

IPv6 enabled or request for LONG format:

```
NETSTAT ROUTE

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
IPv4 Destinations
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UH        000000  OSAQDIOLINK
127.0.0.1/32     0.0.0.0          UH        000002  LOOPBACK
IPv6 Destinations
DestIP:   Default
  Gw:     2001:0db8::206:2aff:fe71:4400
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UGS               MTU:     1492
DestIP:   ::1/128
  Gw:     ::
  Intf:   LOOPBACK6         Refcnt:  000000
  Flgs:   UH                MTU:     65535
DestIP:   2001:0db8::9:67:115:13/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UD                MTU:     1492
DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UHS               MTU:     1492
```

```
NETSTAT ROUTE ADDRTYPE IPV4

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
IPv4 Destinations
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UH        000000  OSAQDIOLINK
127.0.0.1/32     0.0.0.0          UH        000002  LOOPBACK
```

```
NETSTAT ROUTE ADDRTYPE IPV6

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
IPv6 Destinations
DestIP:   Default
  Gw:     2001:0db8::206:2aff:fe71:4400
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UGS               MTU:     1492
DestIP:   ::1/128
  Gw:     ::
  Intf:   LOOPBACK6         Refcnt:  000000
  Flgs:   UH                MTU:     65535
DestIP:   2001:0db8::9:67:115:13/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UD                MTU:     1492
DestIP:   2001:0db8::206:2aff:fe71:4400/128
  Gw:     ::
  Intf:   OSAQDIO46         Refcnt:  000000
  Flgs:   UHS               MTU:     1492
```

```
NETSTAT ROUTE DETAIL

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:03:13
IPv4 Destinations
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.1       UGS       000000  OSAQDIO5L
  Metric: 00000001
  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:  000000
  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 V1R9       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
```

```
NETSTAT ROUTE IQDIO

MVS TCP/IP NETSTAT CS V1R9       TCPIP NAME: TCPCS            09:51:02
Destination      Gateway          Interface
-----------      -------          ---------
9.67.1.9/32      0.0.0.0          LIQDIO1
```

```
NETSTAT ROUTE PR prtable1

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           14:24:09
IPv4 Destinations
Policy Routing Table: prtable1
  IgnorePathMtuUpdate: Yes  MultiPath: Conn(Policy)
  DynamicXCFRoutes:    No
Destination      Gateway          Flags     Refcnt  Interface
-----------      -------          -----     ------  ---------
Default          9.67.115.65      UGS       000002  OSAQDIOLINK
9.67.115.65/32   0.0.0.0          UHS       000000  OSAQDIOLINK
9.67.115.69/32   0.0.0.0          UH        000000  OSAQDIOLINK
```

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.

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 not defined in the stack.

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 will be 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).

Reference count (RefCnt)
The current number of active users for the route.

Interface or Intf
The link or interface name for the route.

MTU 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 is dropped and an ICMP error is returned to the originator of the packet. If a route is inactive, the configured MTU value that was defined using the MTU parameter in the ROUTE statement (or the default MTU value for the specified interface type) is displayed. If a route is active, then the actual MTU value is displayed.

Metric 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 best route or 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.

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 acknowledgements 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

Indicates whether IPv4 ICMP fragmentation-needed messages are ignored for this route 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 route table.

No IPv4 ICMP Fragmentation Needed messages are processed for this route table.

MultiPath

The information in this field is divided into two parts. The value before the parentheses indicates whether the multipath routing selection algorithm for outbound IP traffic is enabled for this policy-based route 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 traffic uses the round-robin distribution method to use multipath routes for each outbound packet.

Conn Indicates that outbound traffic uses the round-robin distribution method to use multipath routes for each outbound connection request.

No Indicates that outbound traffic always uses the first active route in a multipath group.

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*.

DynamicXCFRoutes

Indicates whether direct routes to the dynamic XCF addresses on other TCP/IP stacks are added to the policy-based route table when the dynamic XCF links to those stacks are active. These are the same routes that are automatically generated in the main route table when the 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:

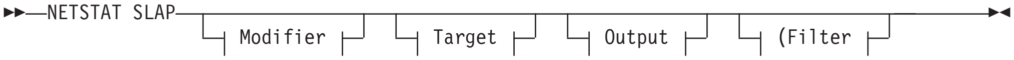
Yes Direct routes to the dynamic XCF addresses on other TCP/IP stacks are added to the policy-based route table when the dynamic XCF links to those stacks are active.

No Direct routes to the dynamic XCF addresses on other TCP/IP stacks are not added to the policy-based route table when the dynamic XCF links to those stacks are active.

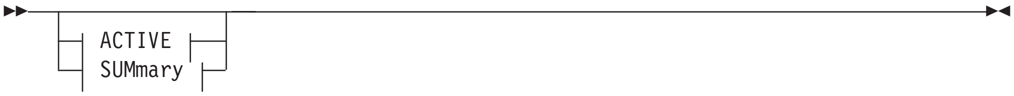
Netstat SLAP/-j report

Purpose: 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:



Modifier:



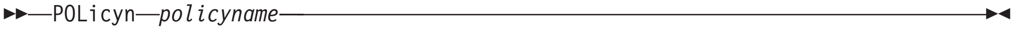
ACTIVE
Display QoS policy information only for the activated policies.

SUMmary
Display a summary of QoS policy information.

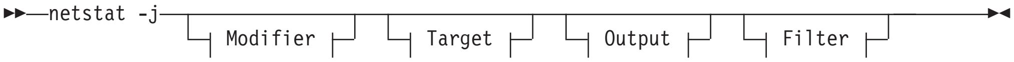
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

Filter:



z/OS UNIX syntax:



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 "Target" on page 263 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 256 or "Output" on page 263.

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, care should be taken 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, the character string should be surrounded by 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 V1R9       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 V1R9       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 will be 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

Purpose: Displays information about each client using a socket application programming interface.

When you specify the NETSTAT SOCKets command, information about the client using a socket application programming interface is displayed along with information about the sockets and associated connections owned by the client.

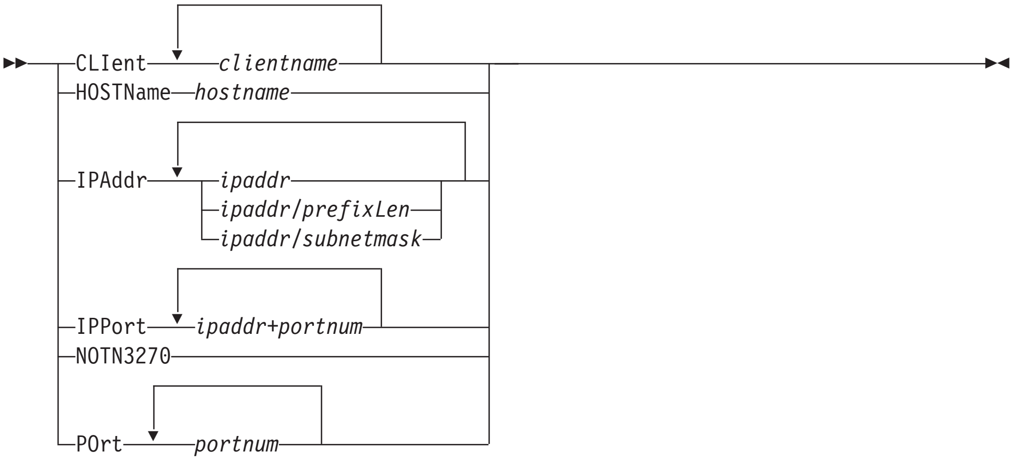
TSO syntax:

```
►►──NETSTAT SOCKets──┬──────────┬──┬──────────┬──┬──────────┬──────────►◄
                     └┤ Target ├┘  └┤ Output ├┘  └┤ (Filter ├┘
```

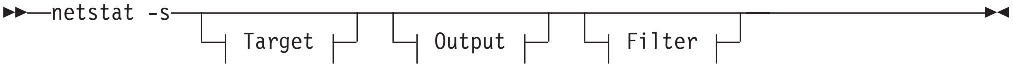
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

Filter:



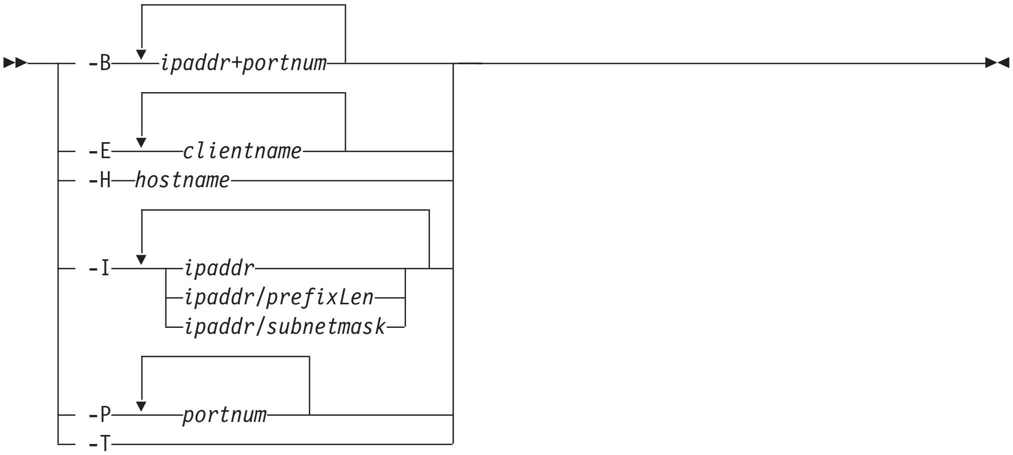
z/OS UNIX syntax:



Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

Filter:



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 256 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.

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 – 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.

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.

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, care should be taken 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, the character string should be surrounded by 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 V1R9       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 V1R9       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 is 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 **internet address..bound port** where internet 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 269 for more information.

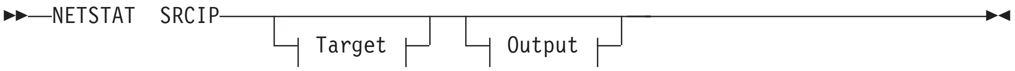
Conn Displays the client identifier, which is a unique number assigned by the TCP/UDP stack to uniquely identify a socket entity.

Netstat SRCIP/-J report

Purpose: Displays the configured information for all job-specific and destination-specific source IP address or interface designations on the TCP/IP address space.

The job-specific and destination-specific source IP address or interface associations displayed by this report are established using the SRCIP profile statement. See *z/OS Communications Server: IP Configuration Reference* for more information about the SRCIP statement.

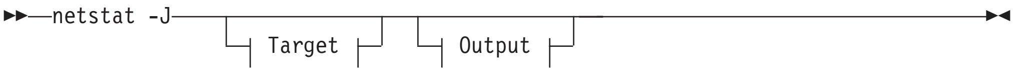
TSO syntax:



Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 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 251 or “Output” on page 263.

z/OS UNIX syntax:



Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 V1R9       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 V1R9       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
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 or jobs for which the designated interface should be used as the source IP address. The job name can end in an asterisk (*), and any actual executing job name that begins with the same characters that precede the asterisk will match this designation. If several different designations exist, then the actual source IP address used will be determined by the most complete match - either an exact match, or the most characters before the asterisk in the job-specific source IP address designation. The job name of asterisk (*) indicates that all the applications that issue TCP connect requests are associated with the specified source IP address or interface, overriding any existing TCPSTACKSOURCEVIPA specifications except for outbound connections whose destination address matches a Destination value.

Type The address family to which this job-specific source IP address applies, either IPv4 or 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).

Source

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).

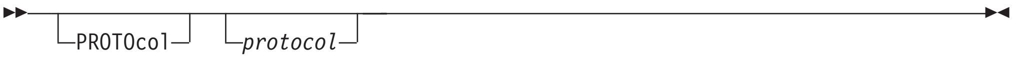
Netstat STATS/-S report

Purpose: 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:

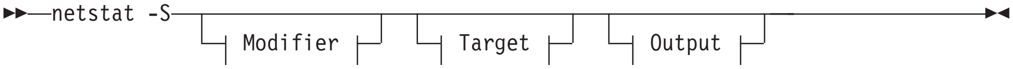


PROTOcol *protocol*
: Display statistics for the specified protocol. The valid protocols are IP, ICMP, TCP, and UDP.

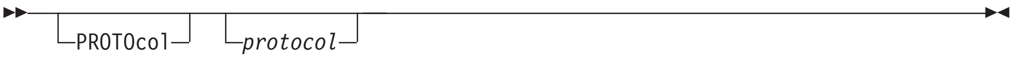
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

z/OS UNIX syntax:



Modifier:



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

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 is 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 is also displayed.
NETSTAT IDS PROTOCOL TCP
   Provides TCP/IP statistics for TCP protocol.
NETSTAT IDS 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 V1R9       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
  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 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
UDP Statistics
  Datagrams Received    = 6984
  No Port Errors        = 2312
  Receive Errors        = 0
  Datagrams Sent        = 368
```

```
NETSTAT STATS PROTOCOL IP

MVS TCP/IP NETSTAT CS V1R9       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 V1R9       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 V1R9       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
  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 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
UDP Statistics
  Datagrams Received    = 0
  No Port Errors        = 12
  Receive Errors        = 0
  Datagrams Sent        = 12
```

```
NETSTAT STATS PROTOCOL IP

MVS TCP/IP NETSTAT CS V1R9       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 V1R9       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. The Packets Received counter includes the packets 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. 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.

  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.

- 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.

Active Connections Opened
The number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state.

Passive Connections Opened
The number of times TCP connections have made a direct transition to the SYN-RCVD state from the LISTEN state.

Connections Closed
Number of TCP connections that have corresponding sockets closed.

Established Connections Dropped
The number of times TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.

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.

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.

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.

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.

Connections Dropped by KeepAlive
Number of TCP connections dropped due to no response while sending keepalive probe requests.

Connections Dropped by Finwait2
Number of TCP connections dropped due to FINWAIT2 timer expiring prior to receiving FIN segments.

- 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.

Netstat TELnet/-t report

Purpose: Displays information for TN3270E Telnet server connections.

TSO syntax:

```
►►──NETSTAT Telnet──┬──────────────┬──┬────────────┬──┬────────────┬──┬─────────────┬──────►◄
                    └─┤ Modifier ├─┘  └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ (Filter ├─┘
```

Modifier:

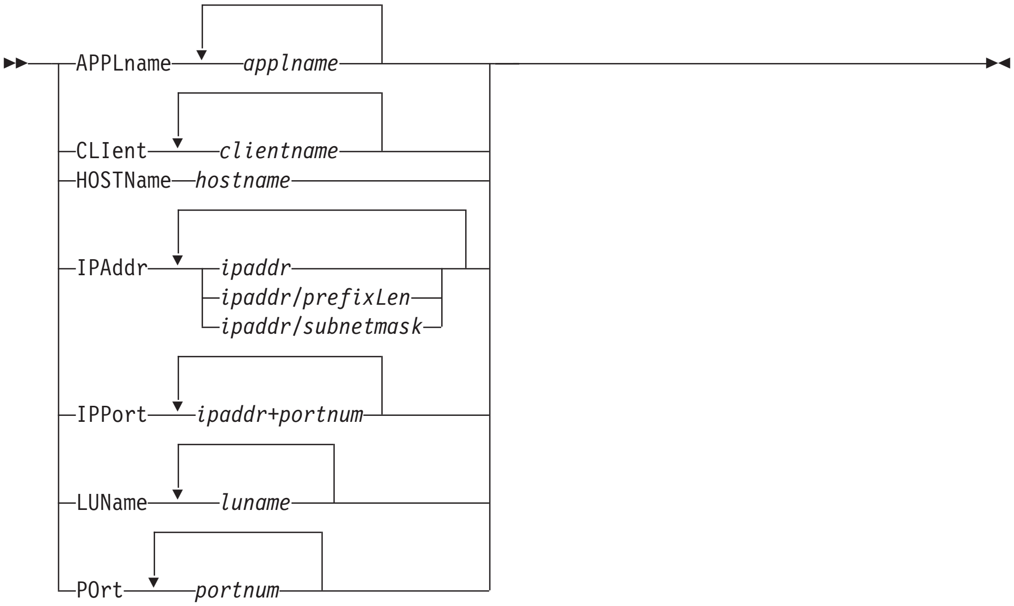
```
►►──DETAIL──────────────────────────────────────────────────────────────►◄
```

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.

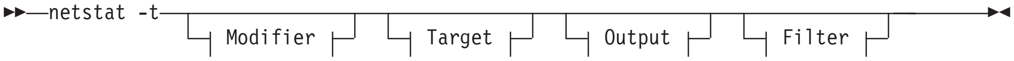
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

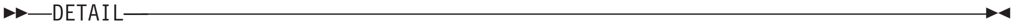
Filter:



z/OS UNIX syntax:



Modifier:



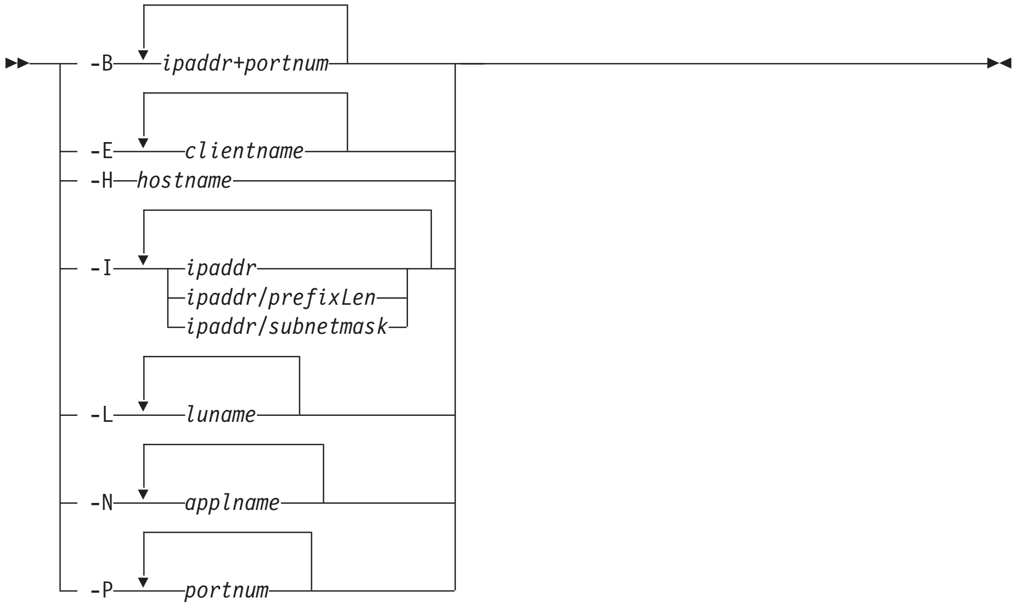
DETAIL

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 -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

Filter:



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 256 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, care should be taken 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, the character string should be surrounded by 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 V1R9       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 V1R9       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 V1R9       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 V1R9       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 1 000)

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 "General concepts" on page 269 for a detailed description.

Foreign Socket
: See the Foreign Socket information in "General concepts" on page 269 for a detailed description.

State The connection state as it is known to TCP/IP. See the TCP connection status information in "General concepts" on page 269 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

Purpose: 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.

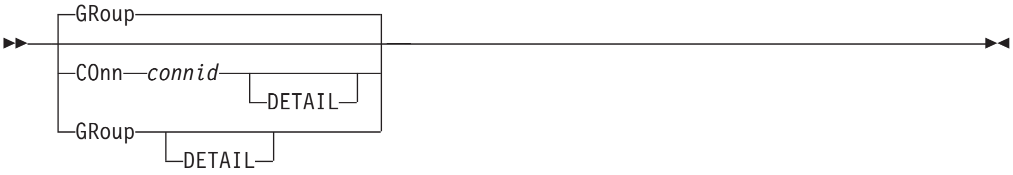
Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

z/OS UNIX syntax:

```
►►──netstat── -x──┬────────────┬──┬──────────┬──┬──────────┬──────────►◄
                  └┤ Modifier ├┘  └┤ Target ├┘  └┤ Output ├┘
```

Modifier:



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 “Target” on page 263 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 251 or “Output” on page 263.

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
```

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 TTLS CONN 1B

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           12:55:20
ConnID: 0000001B
  JobName:      THISJOB
  LocalSocket:  9.67.103.6..72
  RemoteSocket: 9.27.11.182..4665
  SecLevel:     TLS Version 1
  Cipher:       0A  TLS_RSA_WITH_3DES_EDE_CBC_SHA
  CertUserID:   THISUSER
  MapType:      Primary
TTLSRule: TTLSRule5
  TTLSGrpAction:  TTLSGrpAction1
  TTLSEnvAction:  TTLSEnvAction1 (Stale)
  TTLSConnAction: TTLSConnAction6
```

```
NETSTAT TTLS CONN 1B DETAIL

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           12:55:20
ConnID: 0000001B
  JobName:      THISJOB
  LocalSocket:  9.67.103.6..72
  RemoteSocket: 9.27.11.182..4665
  SecLevel:     TLS Version 1
  Cipher:       0A  TLS_RSA_WITH_3DES_EDE_CBC_SHA
  CertUserID:   THISUSER
  MapType:      Primary
TTLSRule: TTLSRule5
  Priority:       2
  LocalAddr:      9.67.103.0/24
  LocalPortFrom:  72                LocalPortTo:  72
  RemoteAddr:     9.27.11/24
  RemotePortFrom: 4000              RemotePortTo: 5000
  JobName:        THIS*
  UserID:         THATUSER
  Direction:      Inbound
  TTLSGrpAction:  TTLSGrpAction1
    GroupID:                    3
    GroupUserInstance:          4
    TTLSEnabled:                On
    Envfile:                    /etc/ttls/group1.env
    CtraceClearText:            Off
    Trace:                      2
    SyslogFacility:             Daemon
    SecondaryMap:               Off
  TTLSEnvAction:  TTLSEnvAction1 (Stale)
    EnvironmentUserInstance:    2
    HandshakeRole:              Client
    Keyring:                    /gsk/rngfil1.rng
    KeyringPw:                  Yes
    KeyringStashFile:           /gsk/stshfil1.sth
    V2CipherSuites:             5 TLS_IDEA_128_CBC_WITH_MD5
                                1 TLS_RC4_128_WITH_MD5
                                3 TLS_RC2_CBC_128_CBC_WITH_MD5
    V3CipherSuites:             0C TLS_DH_DSS_WITH_DES_CBC_SHA
                                0F TLS_DH_RSA_WITH_DES_CBC_SHA
                                36 TLS_DH_DSS_WITH_AES_256_CBC_SHA
    CtraceClearText:            On
    Trace:                      14
    SSLV2:                      Off
    SSLV3:                      Off
    TLSV1:                      On
    ResetCipherTimer:           600
    ApplicationControlled:      Off
    ClientAuthType:             Required
    HandshakeTimeout:           5
    CertificateLabel:           RINGLBL7
    ClientAuthType:             Required
    SecondaryMap:               Off
    GSK_V2_SESSION_TIMEOUT:     3000
    GSK_V2_SIDCACHE_SIZE:       16000
    GSK_V3_SESSION_TIMEOUT:     7000
    GSK_V3_SIDCACHE_SIZE:       32000
    GSK_SYSPLEX_SIDCACHE:       On
    GSK_LDAP_SERVER:            SERVER.NAME
    GSK_LDAP_SERVER_PORT:       976
    GSK_LDAP_USER:              LDAPUSR1
    GSK_LDAP_USER_PW:           No
    GSK_CRL_CACHE_TIMEOUT:      500
```

```
TTLSConnAction: TTLSConnAction6
  ConnectionUserInstance:     2
  HandshakeRole:              Server
  V2CipherSuites:             6 TLS_DES_64_CBC_WITH_MD5
                              1 TLS_RC4_128_WITH_MD5
                              3 TLS_RC2_CBC_128_CBC_WITH_MD5
  V3CipherSuites:             0A TLS_RSA_WITH_3DES_EDE_CBC_SHA
                              0F TLS_DH_RSA_WITH_DES_CBC_SHA
                              36 TLS_DH_DSS_WITH_AES_256_CBC_SHA
  CtraceClearText:            On
  Trace:                      14
  SSLV2:                      Off
  SSLV3:                      On
  TLSV1:                      Off
  ResetCipherTimer:           600
  ApplicationControlled:      Off
  HandshakeTimeout:           10
  CertificateLabel:           RINGLBL6
  SecondaryMap:               Off
```

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, if specified, overrides the CertificateLabel value on the TTLSEnvironmentAction. If CertificateLabel is not specified on either the TTLSConnectionAction or the TTLSEnvironmentAction, the keyring default certificate is used.

CertUserID
The user ID, if any, associated with the partner's certificate. If no associated user ID is available, N/A is displayed.

Cipher
The cipher currently in use for encryption/decryption of data for the connection.

ClientAuthType
The level of Client Authentication used when the HandshakeRole is set to a value of ServerWithClientAuth. The following are 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 isn't 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.

ConnID

The TCP/IP stack defined unique connection ID representing the connection.

ConnectionUserInstance

The instance identifier configured for the TTLSConnectionAction 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.

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, if specified, overrides the CtraceClearText value on the TTLSEnvironmentAction which, in turn, (if specified) overrides the CtraceClearText value on the TTLSGroupAction.

Direction

The connection direction condition specified in the policy rule that was mapped to the connection. The following are valid 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 configured for the TTLSEnvironmentAction 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.

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 configured for the TTLSGroupAction 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 will remain valid. A value of 0 indicates that CRL caching is disabled. See *z/OS Cryptographic Service System Secure Sockets Layer Programming* for details.

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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 Service System Secure Sockets Layer 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 – 60.

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. The following are valid 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.

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 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 will match 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 will match 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 "General concepts" on page 269 for a detailed description.

MapType
The mapping method used to locate this policy. The following are valid 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. 0 is the default value.

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 will match 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 will match 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 "General concepts" on page 269 for a detailed description.

ResetCipherTimer
The number of minutes a secure connection can be active before a rehandshake is performed to establish a new session key for the connection. If not specified, cipher reset is not performed. Valid values are in the range 0 – 1440.

Result: For a particular connection the ResetCipherTimer value on the TTLSConnectionAction, if specified, overrides the ResetCipherTimer value on the TTLSEnvironmentAction.

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.

SSLV2 Indicates whether SSL version 2 protocol is acceptable for the connection.

Result: For a particular connection the SSLV2 value on the TTLSConnectionAction, if specified, overrides the SSLV2 value on the TTLSEnvironmentAction.

SSLV3 Indicates whether SSL version 3 protocol is acceptable for the connection.

Result: For a particular connection the SSLV3 value on the TTLSConnectionAction, if specified, overrides the SSLV3 value on the TTLSEnvironmentAction.

SyslogFacility
The syslog facility name this group uses when writing records to syslogd.

TLSV1
Indicates whether TLS version 1 protocol is acceptable for the connection.

Result: For a particular connection the TLSV1 value on the TTLSConnectionAction, if specified, overrides the TLSV1 value on the TTLSEnvironmentAction.

TTLSConnAction
The name of the policy action used to specify attribute differences between what is desired 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 will be 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 which in turn, if specified, overrides the Trace value on the TTLSGroupAction.

The level of tracing is a sum of the following numbers:

0	No tracing is enabled.
1	Error - Errors are traced to the TCPIP joblog.
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 will trace 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

mywq70gsk_environment_open() in *z/OS Cryptographic Service System Secure Sockets Layer Programming* for a list of valid cipher specifications.

Result: For a particular connection the V2CipherSuites value on the TTLSConnectionAction, if specified, overrides the V2CipherSuites value on the TTLSEnvironmentAction.

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. See gsk_environment_open() in *z/OS Cryptographic Service System Secure Sockets Layer Programming* for a list of valid cipher specifications.

Result: For a particular connection the V3CipherSuites value on the TTLSConnectionAction, if specified, overrides the V3CipherSuites value on the TTLSEnvironmentAction.

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.

Report examples:

NETSTAT TTLS GROUP

```
MVS TCP/IP NETSTAT CS V1R9       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 V1R9      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

Purpose: 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 “Target” on page 263 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 251 or “Output” on page 263.

z/OS UNIX syntax:

```
►►──netstat -u──┬──────────┬─┬──────────┬─────────────────────────►◄
                └┤ Target ├┘ └┤ Output ├┘
```

Target: Provide the report for a specific TCP/IP address space by using -p *tcpname*. See “Target” on page 263 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 256 or “Output” on page 263.

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 V1R9       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 V1R9       TCPIP Name: TCPCS           14:34:37
Tcpip started at 14:27:29 on 01/31/2002 with IPv6 enabled
```

Netstat VCRT/-V report

Purpose: 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.
- Information about the route used by the stack which owns a dynamic VIPA to send packets to the target stack. This information will not be 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.

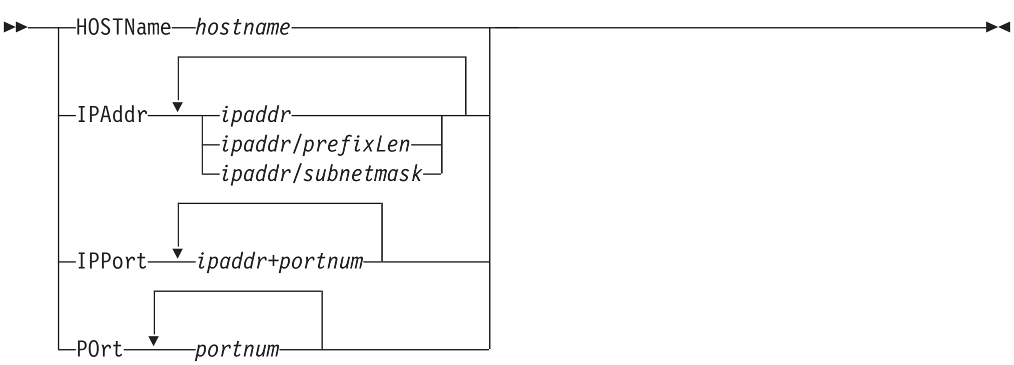
  The routing information might describe the best available route to reach the IP address, which was defined in the VIPAROUTE statement for that target stack, or it might describe the dynamic XCF route for that target stack. See VIPADYNAMIC in the *z/OS Communications Server: IP*

Configuration Reference for information about the VIPAROUTE statement. For more information about the use of the routing information, see Route selection for distributing packets in the *z/OS Communications Server: IP Configuration Guide*.

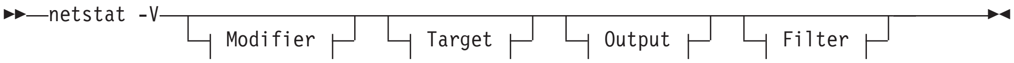
Target: Provides the report for a specific TCP/IP address space by using TCp *tcpname*. See “Target” on page 263 for more information about the TCp parameter.

Output: The default output option displays the output on the user’s terminal. For other options, see “The TSO NETSTAT command syntax” on page 251 or “Output” on page 263.

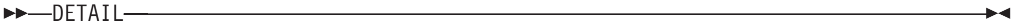
Filter:



z/OS UNIX syntax:



Modifier:



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.
- Information about the route used by the stack which owns a dynamic VIPA to send packets to the target stack. This information will not be 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.

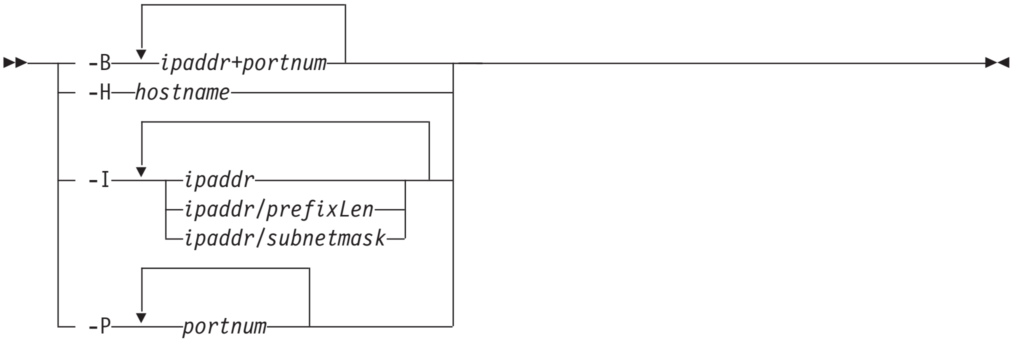
  The routing information might describe the best available route to reach the IP address, which was defined in the VIPAROUTE statement for that target stack, or it might describe the dynamic XCF route for that target stack. See VIPADYNAMIC in the *z/OS Communications Server: IP Configuration Reference* for information about the VIPAROUTE statement.

For more information about the use of the routing information, see 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 “Target” on page 263 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 256 or “Output” on page 263.

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 256 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 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.

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 – 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.

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.

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 V1R9       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 V1R9       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
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
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
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
```

IPv6 enabled or request for LONG format:

```
NETSTAT VCRT

MVS TCP/IP NETSTAT CS V1R9       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 V1R9       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
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
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
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

 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
Dynamic XCF address of stack processing this connection.

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
Dynamic XCF address of stack processing this connection.

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. A 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 TCPIP 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 in the *z/OS Communications Server: IP Configuration Reference* for information 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.

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

Purpose: Displays the dynamic VIPA destination port table information. The destination port table exists only on distributing stacks. These are stacks where a VIPADISTRIBUTE DEFINE 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 “Target” on page 263 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 251 or “Output” on page 263.

Filter:

```
                       ┌───────────────────────┐
                       ▼                       │
►►──┬──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 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 "Target" on page 263 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 256 or "Output" on page 263.

Filter:

```
             ┌───────────────┐
             ▼               │
►►──┬─ -B─────ipaddr+portnum─┴───────┬────────────────────────►◄
    │        ┌────────────────────┐  │
    │        ▼                    │  │
    ├─ -I─────┬─ipaddr────────────┬┴─┤
    │         ├─ipaddr/prefixLen──┤  │
    │         └─ipaddr/subnetmask─┘  │
    │        ┌─────────────┐         │
    │        ▼             │         │
    └─ -P─────────portnum──┴─────────┘
```

Filter description:

HOSTName/-H *hostname*

Filter the output of the VDPT/-O report using the specified host name *hostname*. You can enter one filter value at a time and the specified value can be up to 256 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 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 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.

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 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.

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.

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 V1R9       TCPIP Name: TCPCS           15:35:26
Dynamic VIPA Destination Port Table:
Dest IPaddr     DPort DestXCF Addr    Rdy TotalConn  WLM TSR  Flg
-----------     ----- ------------    --- ---------  --- ---  ---
201.2.10.11     00021 201.1.10.15     001 0000310485 01  075  DRI
201.2.10.12     04011 201.1.10.15     001 0000103162 01  075  R
201.2.10.12     04011 201.2.10.10     001 0000102658 01  050  RI
201.2.10.13     00243 201.3.10.16     001 0000256794 03  085  B
201.2.10.14     00244 201.3.10.16     000 0000000000 15  100  S
201.2.10.15     05000 201.3.10.15     001 0000034011 10  100  A
201.2.10.15     05000 201.3.10.16     002 0000060340 20  100  A
```

```
NETSTAT VDPT DETAIL

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           15:35:26
Dynamic VIPA Distribution Port Table:
Dest IPaddr     DPort DestXCF Addr    Rdy TotalConn  WLM TSR  Flg
-----------     ----- ------------    --- ---------  --- ---  ---
201.2.10.11     00021 201.1.10.15     001 0000310485 01  075  DRI
  TCSR: 100 CER: 075 SEF: 075
  ActConn:      00000042
201.2.10.12     04011 201.1.10.15     001 0000103162 01  075  R
  TCSR: 100 CER: 100 SEF: 075
  ActConn:      00000002
201.2.10.12     04011 201.2.10.10     001 0000102658 01  050  RI
  TCSR: 067 CER: 075 SEF: 075
  ActConn:      00000834
201.2.10.13     00243 201.3.10.16     001 0000256794 03  090  B
  TCSR: 100 CER: 095 SEF: 090
  Weight: 12
    Raw          CP: 13 zAAP: 00 zIIP: 10
    Proportional CP: 08 zAAP: 00 zIIP: 04
  ActConn:      00000091
  QosPlcAct:    *DEFAULT*                                       W/Q:01
201.2.10.14     00244 201.3.10.16     000 0000000000 15  090  S
  TCSR: 100 CER: 095 SEF: 090
  Weight: 60
    Raw          CP: 60 zAAP: 00 zIIP: 60
    Proportional CP: 06 zAAP: 00 zIIP: 54
    ActConn:    00000000
  QosPlcAct:    *DEFAULT*                                       W/Q:01
201.2.10.15     05000 201.3.10.15     001 0000034011 10  100  A
  TCSR: 100 CER: 100 SEF: 100
  Abnorm: 00         Health: 100
  ActConn:      00003011
201.2.10.15     05000 201.3.10.16     002 0000060340 20  100  A
  TCSR: 100 CER: 100 SEF: 100
  Abnorm: 00         Health: 100
  ActConn:      00006003
```

IPv6 enabled or request for LONG format:

```
NETSTAT VDPT

MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           15:37:51
Dynamic VIPA Destination Port Table:
Dest:        201.2.10.11..21
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 075
  Flg: Dynamic, Roundrobin, Inactive
Dest:        201.2.10.12..4011
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 100
  Flg: Roundrobin
Dest:        201.2.10.12..4011
  DestXCF:   201.2.10.10
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 050
  Flg: Roundrobin, Inactive
Dest:        201.2.10.13..243
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 08 TSR: 085
  Flg: BaseWLM
Dest:        201.2.10.14..244
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 15 TSR: 090
  Flg: ServerWLM
Dest:        201.2.10.15..5000
  DestXCF:   201.3.10.15
  TotalConn: 0000034011  Rdy: 001  WLM: 10 TSR: 100
  Flg: WeightedActive
Dest:        201.2.10.15..5000
  DestXCF:   201.3.10.16
  TotalConn: 0000060340  Rdy: 002  WLM: 20 TSR: 100
  Flg: WeightedActive
DestIntf:
  Dest:        2001:0db8::522:f103..20
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 094
    Flg: BaseWLM
DestIntf:
  Dest:        2001:0db8::522:f103..21
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 15 TSR: 100
    Flg: ServerWLM
```

NETSTAT VDPT DETAIL

```
MVS TCP/IP NETSTAT CS V1R9       TCPIP Name: TCPCS           15:37:51
Dynamic VIPA Destination Port Table:
Dest:        201.2.10.11..21
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 075
  Flg: Dynamic, Roundrobin, Inactive
    TCSR: 100 CER: 075 SEF: 100
    ActConn:   0000000000
Dest:        201.2.10.12..4011
  DestXCF:   201.1.10.15
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 100
  Flg: Roundrobin
    TCSR: 100 CER: 100 SEF: 100
    ActConn:   0000000000
Dest:        201.2.10.12..4011
  DestXCF:   201.2.10.10
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 050
  Flg: Roundrobin, Inactive
    TCSR: 067 CER: 075 SEF: 075
    ActConn:   0000000000
Dest:        201.2.10.13..243
  DestXCF:   201.3.10.16
  TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 090
  Flg: BaseWLM
    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
  Flg: ServerWLM
    TCSR: 100 CER: 095 SEF: 090
     Weight: 60
       Raw          CP: 50 zAAP: 00 zIIP: 61
       Proportional CP: 05 zAAP: 00 zIIP: 55
    ActConn:   00000000
    QosPlcAct: *DEFAULT*  W/Q:01
    QosPlcAct: *DEFAULT*
      W/Q: 02
Dest: 201.2.10.15..5000
  DestXCF: 201.3.10.15
  TotalConn: 0000034011 Rdy: 001 WLM: 10 TSR: 100
  Flg: WeightedActive
    TCSR: 100 CER: 100 SEF: 100
    Abnorm: 00         Health: 100
    ActConn:      00003011
Dest: 201.2.10.15..5000
  DestXCF: 201.3.10.16
  TotalConn: 0000060340 Rdy: 002 WLM: 20 TSR: 100
  Flg: WeightedActive
    TCSR: 100 CER: 100 SEF: 100
    Abnorm: 00         Health: 100
    ActConn:      00006003
DestIntf:
  Dest:        2001:0db8::522:f103..20
    DestXCF:   2001:0db8::943:f003
    TotalConn: 0000000000  Rdy: 001  WLM: 01 TSR: 096
    Flg: BaseWLM
    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
    Flg: ServerWLM
    TCSR: 100 CER: 100 SEF: 100
    Weight: 50
      Raw          CP: 60 zAAP: 00 zIIP: 49
      Proportional CP: 06 zAAP: 00 zIIP: 44
    Abnorm:00          Health: 100
    ActConn:   0000000000
    QosPlcAct: *DEFAULT*
      W/Q: 01
```

Report field descriptions:

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
Dynamic XCF address of target stack to receive connections.

Flg Flags; can have one of the following values:

D Indicates a dynamically assigned destination/port entry.

B Indicates that WLM system weights and policy information are used to distribute incoming connection requests.

R Indicates that incoming connection requests are distributed in round-robin method.

I Indicates that the data path to the target stack is inactive.

S 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/Port and all target servers are able to provide WLM server-specific weights. Otherwise, BaseWLM is used.

A Indicates that incoming connection requests are distributed by the weighted active connections method.

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.

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; can have one of the following values:

Dynamic
Indicates a dynamically assigned destination/port entry.

BaseWLM
Indicates that WLM system weights and policy information are used to distribute incoming connection requests.

Roundrobin
Indicates that incoming connection requests are distributed in round-robin method.

Inactive
Indicates that the datapath to the XCF target is inactive.

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/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 by the weighted active connections method.

Local Indicates that the target stack specified by the DestXCF Addr value is currently processing outbound connections for this destination and port pair locally.

For a SHORT or LONG format report:

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.

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.

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 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.

- A value of 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.
- A value of 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 lower of the server accept efficiency fraction (SEF) and connection establishment rate (CER).

- The TCSR measures the percentage of connection setup requests routed from the distributor that are successfully received by the target for this server.
- The CER is a subcomponent of the SEF and measures the percentage of the connection setup requests received by the target for this server that achieve the connection established state.
- 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.

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.

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 a subcomponent of the SEF value. For information about diagnosing sysplex problems, see the steps for diagnosing sysplex problems in the *z/OS Communications Server: IP Diagnosis Guide*.

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 should be 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:

- 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.

Netstat VIPADCFG/-F report

Purpose: 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:

- 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.

Target: Provide the report for a specific TCP/IP address space by using TCp *tcpname*. See "Target" on page 263 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 251 or "Output" on page 263.

Filter:

```
          ┌────────────────────────┐
          ▼                        │
►►──IPAddr──┬─ipaddr──────────────┬┴──────────────────────────────────────►◄
            ├─ipaddr/prefixLen────┤
            └─ipaddr/subnetmask───┘
```

z/OS UNIX syntax:

```
►►──netstat -F──┬──────────────┬──┬────────────┬──┬────────────┬──┬────────────┬──────►◄
                └─┤ Modifier ├─┘  └─┤ Target ├─┘  └─┤ Output ├─┘  └─┤ (Filter ├─┘
```

Modifier:

```
►►──DETAIL──────────────────────────────────────────────────────────────────►◄
```

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 "Target" on page 263 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 256 or "Output" on page 263.

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 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 V1R9       TCPIP Name: TCPCS           19:47:49
Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr
   ----------       ----    ------------     --------  ------
   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

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr
   ----------       -----------      --------  ------
   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

 VIPA Range:
   AddressMask      IP Address       Moveable
   -----------      ----------       --------
   255.255.255.192  201.2.10.192     NonDisr
   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
   201.2.10.12      4011   ALL              Yes    No      R
   201.2.10.13      243    ALL              No     No      B
   201.2.10.14      243    ALL              No     No      S
   201.2.10.15      5000   201.3.10.15      No     No      A
   201.2.10.15      5000   201.3.10.16      No     No      A

 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
   ----------       ----    ------------     --------  ------
   201.2.10.40      100     255.255.255.192  Immediate No

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr
   ----------       -----------      --------  ------
   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 V1R9       TCPIP Name: TCPCS           19:47:49
Dynamic VIPA Information:

 VIPA Backup:
   IP Address       Rank    Address Mask     Moveable  SrvMgr
   ----------       ----    ------------     --------  ------
   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

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr
   ----------       -----------      --------  ------
   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

 VIPA Range:
   AddressMask      IP Address       Moveable
   -----------      ----------       --------
   255.255.255.192  201.2.10.192     NonDisr
   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
     OptLoc: No
   201.2.10.12      4011   ALL              Yes    No      RO
     OptLoc: 1
   201.2.10.13      243    ALL              No     No      B
     OptLoc: No
     ProcType:
       CP: 60  zAAP: 00  zIIP: 40
   201.2.10.14      243    ALL              No     No      S
     OptLoc: No
   201.2.10.15      5000   201.3.10.15      No     No      A
     OptLoc: No     Weight: 10
   201.2.10.15      5000   201.3.10.16      No     No      A
     OptLoc: No     Weight: 20

 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
   ----------       ----    ------------     --------  ------
   201.2.10.40      100     255.255.255.192  Immediate No

 VIPA Define:
   IP Address       AddressMask      Moveable  SrvMgr
   ----------       -----------      --------  ------
   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 V1R9       TCPIP Name: TCPCS           19:49:12
Dynamic VIPA Information:

  VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.29/26
      Rank: 025  Moveable: WhenIdle   SrvMgr: Yes
    IpAddr/PrefixLen: 201.2.10.30/26
      Rank: 025  Moveable: Immediate  SrvMgr: No
    IpAddr/PrefixLen: 201.2.10.32
      Rank: 040  Moveable:            SrvMgr:
    IntfName: INTFNAM6
      IpAddr: 2001:db8::526:f603
        Rank: 050  Moveable:            SrvMgr: n/a

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.11/26
      Moveable: WhenIdle   SrvMgr: No
    IpAddr/PrefixLen: 201.2.10.12/26
      Moveable: Immediate  SrvMgr: Yes
    IpAddr/PrefixLen: 201.2.10.13/26
      Moveable: Immediate  SrvMgr: No
    IpAddr/PrefixLen: 201.2.10.14/26
      Moveable: Immediate SrvMgr: No
    IntfName: INTFNAM1
      IpAddr: 2001:0db8::522:f103
        Moveable: Immediate  SrvMgr: n/a
    IntfName: INTFNAM2
      IpAddr: 2001:0db8::522:f203
        Moveable: Immediate  SrvMgr: n/a

  VIPA Range:
    IpAddr/PrefixLen: 201.2.10.192/26
      Moveable: NonDisr
    IpAddr/PrefixLen: 201.2.20.192/26
      Moveable: Disrupt
    IntfName: INTFNAM3
      IpAddr/PrefixLen: 2001:0db8::522:f303/24
        Moveable: NonDisr

  VIPA Distribute:
    Dest:        201.2.10.11..n/a
      DestXCF:   ALL
        SysPt:   Yes  TimAff: 200  Flg: Roundrobin
    Dest:        201.2.10.12..4011
      DestXCF:   ALL
        SysPt:   Yes  TimAff: No   Flg: Roundrobin
    Dest:        201.2.10.14..243
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: ServerWLM
    Dest:        201.2.10.13..243
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: BaseWLM
    DestIntf:    INTFNAM1
      Dest:      2001:0db8::522:f103..20
        DestXCF: ALL
          SysPt: No   TimAff: No    Flg: ServerWLM
    DestIntf:    INTFNAM1
      Dest:      2001:0db8::522:f103..21
        DestXCF: ALL
          SysPt: Yes  TimAff: 10    Flg: ServerWLM

  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

  VIPA Define:
     IpAddr/PrefixLen: 201.2.10.20/26
      Moveable: Immediate  SrvMgr: No

  VIPA Distribute:
    Dest:        201.2.10.20..5000
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: BaseWLM
```

```
NETSTAT VIPADCFG DETAIL
MVS TCP/IP NETSTAT CS V1R9      TCPIP Name: TCPCS           19:49:12
Dynamic VIPA Information:

  VIPA Backup:
    IpAddr/PrefixLen: 201.2.10.29/26
      Rank: 025  Moveable: WhenIdle   SrvMgr: Yes
    IpAddr/PrefixLen: 201.2.10.30/26
      Rank: 025  Moveable: Immediate  SrvMgr: No
    IpAddr/PrefixLen: 201.2.10.32
      Rank: 040  Moveable:            SrvMgr:
    IntfName: INTFNAM6
      IpAddr: 2001:db8::526:f603
        Rank: 050  Moveable:            SrvMgr: n/a

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.11/26
      Moveable: WhenIdle   SrvMgr: No
    IpAddr/PrefixLen: 201.2.10.12/26
      Moveable: Immediate  SrvMgr: Yes
    IpAddr/PrefixLen: 201.2.10.13/26
      Moveable: Immediate  SrvMgr: No
    IntfName: INTFNAM1
      IpAddr: 2001:0db8::522:f103
        Moveable: Immediate  SrvMgr: n/a
    IntfName: INTFNAM2
      IpAddr: 2001:0db8::522:f203
        Moveable: Immediate  SrvMgr: n/a

  VIPA Range:
    IpAddr/PrefixLen: 201.2.10.192/26
      Moveable: NonDisr
    IpAddr/PrefixLen: 201.2.20.192/26
      Moveable: Disrupt
    IntfName: INTFNAM3
      IpAddr/PrefixLen: 2001:0db8::522:f303/24
        Moveable: NonDisr

  VIPA Distribute:
    Dest:        201.2.10.11..n/a
      DestXCF:   ALL
        SysPt:   Yes  imAff: 200  Flg: Roundrobin
        OptLoc:  No
    Dest:        201.2.10.12..4011
      DestXCF:   ALL
        SysPt:   Yes  TimAff: No   Flg: Roundrobin OptLocal
        OptLoc:  1
    Dest:        201.2.10.13..243
      DestXCF:   ALL
        SysPt:   No   TimAff: No   Flg: BaseWLM
        OptLoc:  No
        ProcType:
          CP: 60  zAAP: 00  zIIP: 40
    Dest:        201.2.10.15..5000
      DestXCF:   201.3.10.15
        SysPt:   No   TimAff: No  Flg: WeightedActive
        OptLoc: No     Weight: 10
    Dest:        201.2.10.15..5000
      DestXCF:   201.3.10.16
        SysPt:   No   TimAff: No  Flg: WeightedActive
        OptLoc: No     Weight: 20
    DestIntf:    INTFNAM1
      Dest:      2001:0db8::522:f103..20
        DestXCF: ALL
          SysPt: No   TimAff: No   Flg: 1ServerWLM
           OptLoc: No    DestIntf:    INTFNAM1
      Dest:      2001:0db8::522:f103..21
        DestXCF: ALL
          SysPt: Yes  TimAff: 10   Flg: 1ServerWLM
           OptLoc: No
  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

  VIPA Define:
    IpAddr/PrefixLen: 201.2.10.20/26
      Moveable: Immediate  SrvMgr: No

  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 Internet address for this DVIPA.

AddressMask

The net mask that determines how many of the bits of the IP address determine the net. This field will be 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 Internet 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 will be 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 will be 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 will be 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.

VIPA Define
Displays the configured dynamic VIPA define information.

For a SHORT format report:

IP Address
The Internet 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 Internet 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 will be 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 will always display **n/a** as it is not applicable for IPv6.

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 Internet 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.

For a LONG format report:

IntfName

The name of this IPv6 interface.

IpAddr/PrefixLen

The Internet 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 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 will move the DVIPA to the new stack, but connections on that DVIPA on the first stack will be 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.

Weight
The configured distribution method is WEIGHTEDActive. This is the configured active connection weight that will be used when incoming connections are distributed to this target stack.

Flg
Flags including the following:

B
Indicates that the DISTMethod BASEWLM parameter was either defined on the VIPADISTRIBUTE profile statement or is in effect by default.

R
Indicates that the DISTMethod ROUNDROBIN parameter was defined on the VIPADISTRIBUTE profile statement.

S
Indicates that the DISTMethod SERVERWLM parameter was defined on the VIPADISTRIBUTE profile statement.

A
Indicates that the DISTMethod WEIGHTEDActive parameter was defined on the VIPADISTRIBUTE profile statement. To see the active connection weight that is being used for this target stack, issue the Netstat VIPADCFG/-F command with the DETAIL keyword.

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.

Flg Flags including the following:

BaseWLM

Indicates that the DISTMethod BASEWLM parameter was either defined on the VIPADISTRIBUTE profile statement or is in effect by default.

Roundrobin

Indicates that the DISTMethod ROUNDROBIN parameter was defined on the VIPADISTRIBUTE profile statement.

WeightedActive

Indicates that the DISTMethod WEIGHTEDActive parameter was defined on the VIPADISTRIBUTE profile statement. To see the active connection weight that is being used for this target stack, issue the Netstat VIPADCFG/-F command with the DETAIL keyword.

ServerWLM

Indicates that the DISTMethod SERVERWLM parameter was defined on the VIPADISTRIBUTE profile statement.

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.

For a SHORT or LONG format report:

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.

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 specified on the OPTLOCAL statement, if one of the following conditions exists, connections will be 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 will be 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.

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 210 for more information about the command.

Netstat VIPADyn/-v report

Purpose: 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 “Target” on page 263 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 251 or “Output” on page 263.

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 "Target" on page 263 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 256 or "Output" on page 263.

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 V1R9       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
  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 V1R9       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
  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 V1R9       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 V1R9       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  DistStat:
     ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1A
   IPADDR/PREFIXLEN: 199.199.199.9/24
     Status:  Active     Origin: VIPARange BIND   DistStat:
     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  DistStat:
       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 V1R9        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  DistStat:
    ActTime: 03/02/2005 16:45:20                 JobName:  JOBTST1A
  IPADDR/PREFIXLEN: 199.199.199.9/24
    Status:  Active     Origin: VIPARange BIND   DistStat:
    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  DistStat:
      ActTime: 03/02/2005 16:45:20                 JobName:   JOBTST6A

NETSTAT VIPADYN VIPAROUTE

MVS TCP/IP NETSTAT CS V1R9        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 Internet 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 Internet 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:

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. Connections on this stack for this DVIPA that were established prior to the move are still being 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 will be removed from this stack when all its connections complete.

Origin
Indicates how the DVIPA was created. It can be one of the following:

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 fell 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:

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.

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 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 desired 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 13. 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 utilizing 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.
DEvlinks	-d	Displays information about devices and defined interfaces or links 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).
PORTList	-o	Displays the reserved port list.
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.

Table 13. z/OS UNIX and TSO Netstat command options (continued)

TSO option	z/OS UNIX option	Description
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.
HOSTName	-H	Filter the output of the ALL/-A, ALLConn/-a, BYTEinfo/-b, COnn/-c, 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.
IPAddr	-I	Filter the output of the ALL/-A, ALLConn/-a, BYTEinfo/-b, COnn/-c, Gate/-g, ND/-n, 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.

Table 13. z/OS UNIX and TSO Netstat command options (continued)

TSO option	z/OS UNIX option	Description
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

Purpose

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 763.

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─┘
```

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 should return 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 should be returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses should be 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 will be 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 of 1 is used. If *echo* is not specified, an error occurs. The *echo* value must be in the range 0 – 2^{31} minus 1, which is 2 147 483 647. If *echo* is 0, the Ping command sends echo requests continually. To stop the Ping command, press PA1 .

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 should be 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.

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 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 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.

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 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 514.

Examples

- IPv4

```
ping mvs098
CS V1R9: Pinging host MVS098 (9.67.113.11)
Ping #1 response took 0.002 seconds.
```

- IPv6

```
ping linuxipv62.tcp
CS V1R9: 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 V1R9: 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 V1R9: 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)
```

- 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 V1R9: Pinging host hostipv6.raleigh.ibm.com (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 V1R9: Pinging host hostipv6.raleigh.ibm.com (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 link-local with scope information.

```
ping fe80::12:1:2%mpc6221
 CS V1R9: Pinging host FE80::12:1:2%MPC6221
 at IPv6 address fe80::12:1:2
 Ping #1 response took 0.028 seconds.
```

The z/OS UNIX ping command—Send an echo request

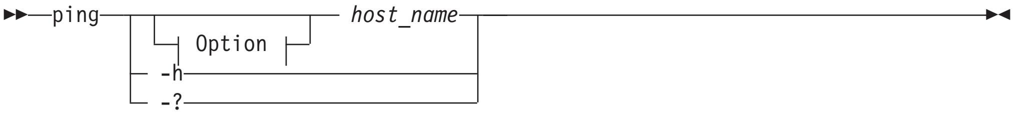
Purpose

The z/OS UNIX **ping** command sends an echo request to a foreign node (remote host) to determine whether the node is accessible.

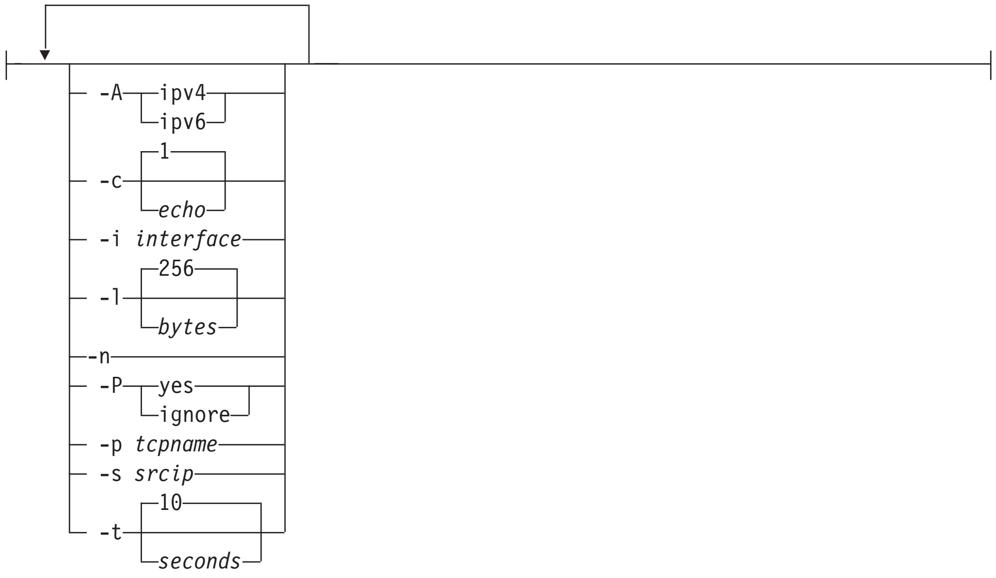
When a response to a Ping command is received, the elapsed time is displayed. The time does not include the time spent communicating between the user and the TCP/IP address space.

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 should return 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 should be returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses should be 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 will be 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 of 1 is used. If *echo* is not specified, an error occurs. The *echo* value must be in the range 0 – 2^{31}–1, which is 2 147 483 647. If *echo* is 0, the Ping command sends echo requests continually. To stop the Ping command, see "Usage" on page 511.

-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 will be 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 will be used to determine whether the host name should be 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.

-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 only in effect 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 will be 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.

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 514.

Examples

- IPv4

```
ping mvs098
CS V1R9: Pinging host mvs098 (9.67.113.11)
Ping #1 response took 0.002 seconds.
```

- IPv6

```
ping linuxipv62.tcp
CS V1R9: 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 V1R9: 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 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.

```
ping -l 2500 -P ignore hosta
CS V1R9: 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: hostk.test.ibm.com (9.56.33.33)
```

- 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 V1R9: Pinging host hostipv6.raleigh.ibm.com (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 V1R9: Pinging host hostipv6.raleigh.ibm.com (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 link-local with scope information.

```
ping fe80::12:1:2%mpc6221
CS V1R9: Pinging host FE80::12:1:2%MPC6221
at IPv6 address fe80::12:1:2
Ping #1 response took 0.001 seconds.
```

TSO PING and z/OS UNIX ping command return codes

The following is a list of the return codes 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. More than one Ping command should be sent 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 should 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

Purpose

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.

Tips:

- 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.

Restrictions:

- 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  portmapper
 100000    2   tcp   111  portmapper
 100003    2   tcp  2049  nfsd
 100003    3   tcp  2049  nfsd
 100003    2   udp  2049  nfsd
 100003    3   udp  2049  nfsd
 100044    1   udp 10001  mvsmount
 100044    1   tcp 10001  mvsmount
 100005    1   udp 10000  mountd
 100005    1   tcp 10000  mountd
```

```
100005    3   udp 10002  mountd
100005    3   tcp 10002  mountd
100059 2 udp 10003 showattrd
100059 2 tcp 10003 showattrd
150001 1 udp 10004 pcnfsd
150001 2 udp 10005 pcnfsd
```

The z/OS UNIX orpcinfo/rpcinfo command—Display server information

Purpose

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.

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, the **rpcinfo** command prints error information. For example, if -u is specified without a versnum value, then 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, **orpcinfo** is used to invoke the nullproc of program 100003 on host mvsx using the TCP protocol. **orpcinfo** invokes the nullproc 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
#
```

Traceroute

The TSO TRACERTE and z/OS UNIX traceroute/otracert commands help you debug network problems.

The TSO TRACERTE command—Debug network problems

Purpose

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 will be increased.

ADDRTYPE

Specifies the IP address type that the Resolver should return 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 should be returned from the Resolver when resolving the host name to an IP address.

ipv4

Specifies that only IPv4 IP addresses should be 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 will be 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 will be 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 will be used to determine whether the host name should be 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.

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 will be 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 only applies to IPv4 destinations and will be 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:

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 should be followed by one of the values below.

FLAG
This is an optional field. It is only present 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 889.

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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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.

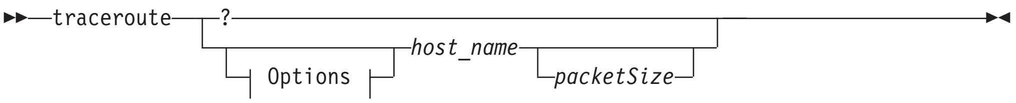
The z/OS UNIX traceroute command—Debug network problems

Purpose

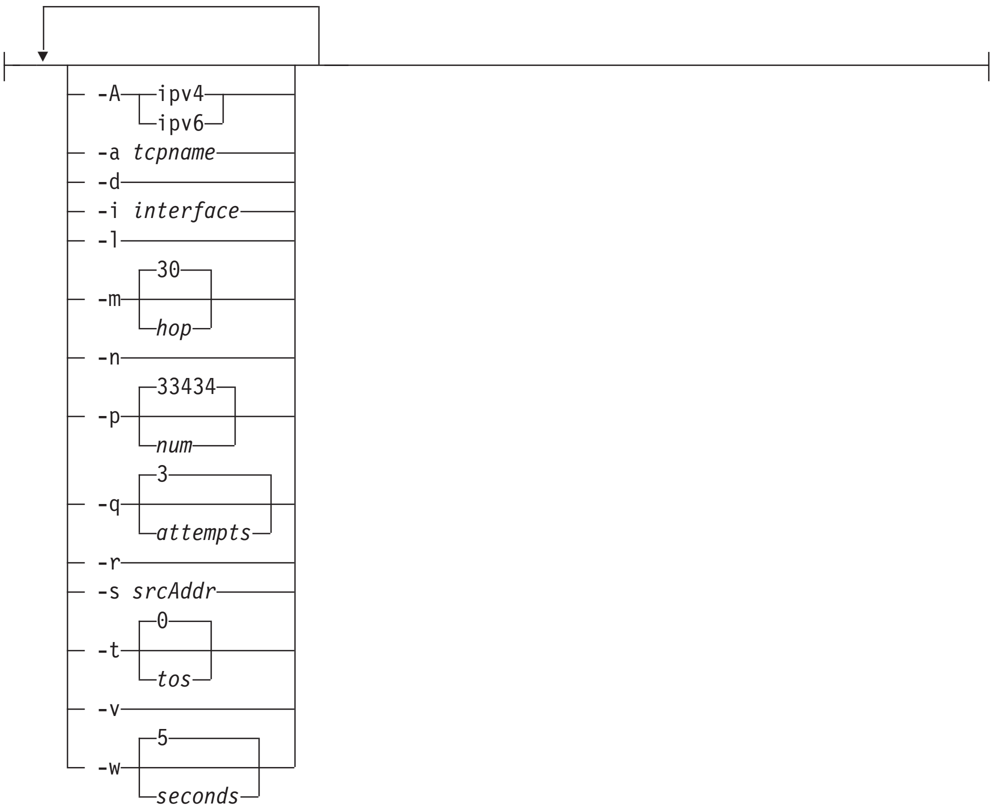
This command is useful for debugging various network problems. This command sends UDP requests with varying TTL (time to live) or hop limit values and then waits for the routers between the local and remote hosts to send time-exceeded messages.

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



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 will be increased.

-A Specifies the IP address type that the Resolver should return 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 should be returned from the Resolver when resolving the host name to an IP address.

ipv4
Specifies that only IPv4 IP addresses should be 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 will be 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 will be used to determine whether the host name should be 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.

-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 only applies to IPv4 destinations and will be 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:

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 should be 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.

FLAG This is an optional field. It is only present 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 889.

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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 V1R9: 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 traceroute uses is normally not valid but can be changed if the target host is using nonstandard UDP port.
- To interrupt traceroute 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 information describes how to use the **ipsec** command to obtain or modify IP security information in the network. The z/OS UNIX **ipsec** command displays and modifies IP security information with respect to a local TCP/IP stack and the IKE daemon or with respect to a network security services (NSS) client that uses the local NSS server's IPSec management service (a TCP/IP stack can be configured as an NSS client by adding a NssStackConfig statement to the configuration file of the stack's IKE daemon.) See the *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.

For additional information that is useful for managing security see the following:

- "MODIFY command—network security services server" on page 142
- "MODIFY command—IKE server" on page 138
- "The z/OS UNIX pasearch command—Display policies" on page 606
- "Netstat CONFIG/-f report" on page 317 shows the IPSECURITY setting defined on the IPCONFIG TCP/IP profile statement
- "Netstat DEvlinks/-d report" on page 345 shows the IPSec security class value as specified on the LINK TCP/IP profile statement for IPv4

Overview of ipsec command

The z/OS UNIX **ipsec** command is used to display and modify IP security information on the local host system or for a NSS client managed by the NSS server. 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
- Display information about the active NSS client configuration
- Display information maintained by the NSS server for each NSS client

The **ipsec** command is an APF-authorized application. Users of the **ipsec** command must also be authorized through the security access facility (SAF). This section assumes the SAF is RACF. You do not have to have 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. See "ipsec command security" on page 538 for details on user authorizations.

As new IP security 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

Users of the **ipsec** command must be authorized through the security access facility. This is managed with the SERVAUTH profile and is described in “SERVAUTH profile.” For the **ipsec -f default** and **ipsec -f reload** command, file system access is also required.

The system administrator might want to allow certain users limited file system access without granting those users full root authority. For details, see “Group access control for local host stacks” on page 540.

SERVAUTH profile

Security product authorization (for example, RACF) is required to use the **ipsec** command. A profile is required 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
```

The following definitions apply:

sysname
The name of the system on which the **ipsec** command is allowed to execute

stackname
The *tcpprocname* value for a local TCP/IP stack

command_type
The **ipsec** command type; either DISPLAY or CONTROL

Resource names in SERVAUTH class	ipsec options
EZB.IPSECCMD.*sysname*.*stackname*.*	All **ipsec** options
EZB.IPSECCMD.*sysname*.*stackname*.DISPLAY	-f display -m display -k display -y display -t -i -o
EZB.IPSECCMD.*sysname*.*stackname*.CONTROL	-f default -f reload -m activate -m deactivate -k deactivate -k refresh -y activate -y deactivate -y refresh

When accessing a remote stack using the NSS server, the following profile format applies:

```
EZB.NETMGMT.sysname.clientname.IPSEC.command_type
```

sysname
The system name on which the **ipsec** command is allowed to execute

clientname
The name of an NSS client

command_type
The **ipsec** command type; either DISPLAY or CONTROL

These profiles must be defined on the system where the NSS server and IPSec 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

Use the following profile format when querying IKED for NSS configuration information using the **ipsec -w** command:

```
EZB.NETMGMT.sysname.sysname.IKED.DISPLAY
```

The following definition applies:

sysname
The name of the system on which the **ipsec** command is allowed to execute

Requirement: This profile must be defined on the system where IKED and IPSec are running.

The format of the profile when accessing the NSS server using the **ipsec -x** command is:

```
EZB.NETMGMT.sysname.sysname.NSS.DISPLAY
```

The following definition applies:

sysname
The name of the system on which the **ipsec** command is allowed to execute

Requirement: This profile must be defined on the system where the NSS server and IPSec 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

In order 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.

Steps for creating group access control over the path of the marker files

A user does not require root authority to use the **ipsec** command, but to avoid erroneous or malicious manipulations of these marker files, the administrator must perform the following steps to require group access control over the path of the marker files.

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. After the supplementary group is created, issue the following UNIX System Services commands to set file management at the group level. The path for file markers is /var/ike.

   ```
   chgrp IKE /var/ike
   chmod 2770 /var/ike
   ```

3. Use RACF commands to control which users can manipulate files at the file marker path (the users that might perform an **ipsec -f default** or **reload** request).

   ```
   CONNECT USER5 GROUP(IKE) UACC(READ)
   REMOVE USER5 GROUP(IKE)
   ```

The z/OS UNIX ipsec command syntax

Purpose

The z/OS UNIX **ipsec** command is used to display and modify IP security information on the host z/OS system. With the -z option or the -x primary option specified, it displays and modifies IP security information for NSS clients using the IPSec management services.

Restriction: When using the **ipsec** command to interface with IPSec management services, the **ipsec** command must be issued on the same host z/OS system as the NSS server.

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 client information using the -w primary option for multiple stacks. It reports NSS server information using the -x primary option for multiple NSS clients. For the other **ipsec** command primary options, the **ipsec** command is always specified for a single stack (using the -p option) or NSS 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.

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.

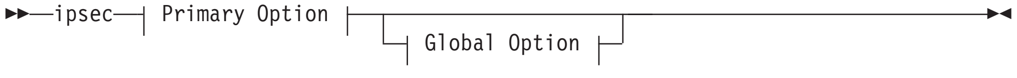
Rule: All policy names are case sensitive.

Tip: Use spaces or commas as valid delimiters to separate **ipsec** command parameter values.

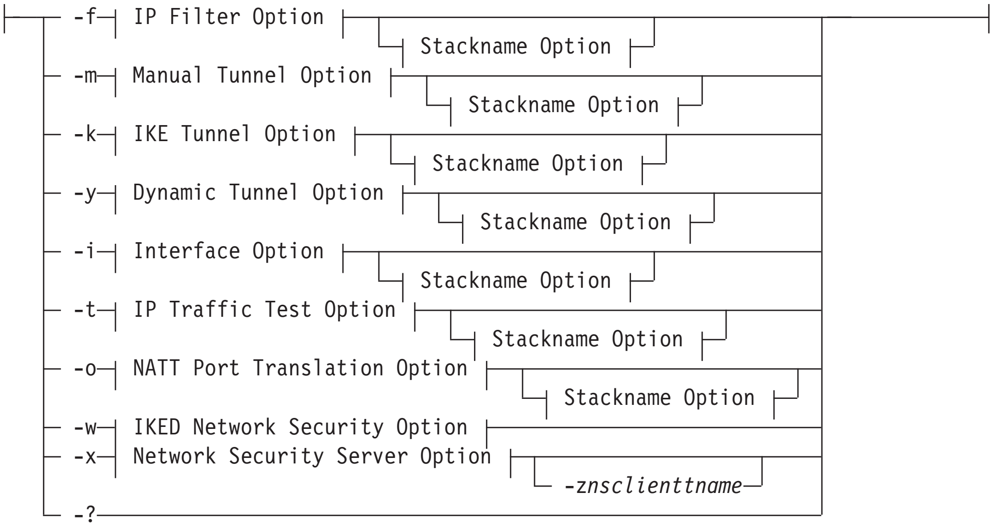
Additionally, as tunnels are initiated and established, they also receive a system-assigned name, known as a tunnel ID. System-assigned tunnel IDs take the form of an integer prefixed with a single letter that identifies the tunnel type. The prefix can be M (manual), K (Internet Key Exchange), or Y (dynamic). The integer is based on a 32-bit counter that is incremented at each assignment and wraps at 4,294,967,295. You should 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.

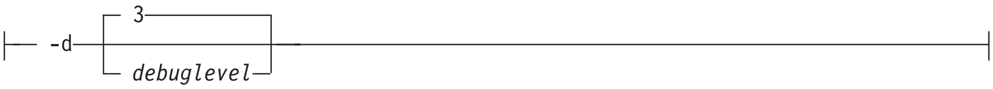
Format



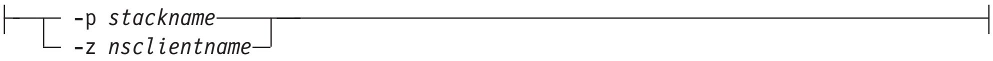
Primary Option:



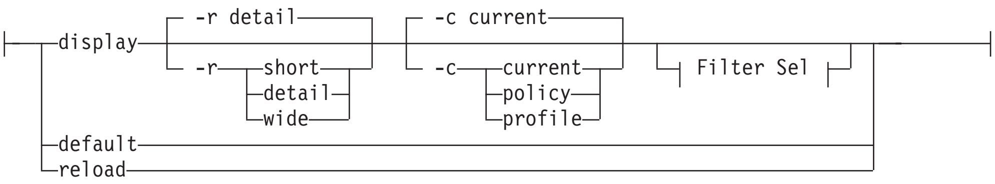
Global Option:



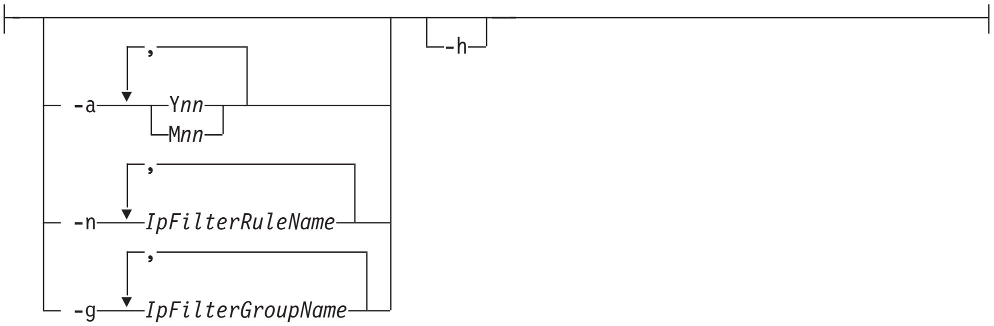
Stackname Option:



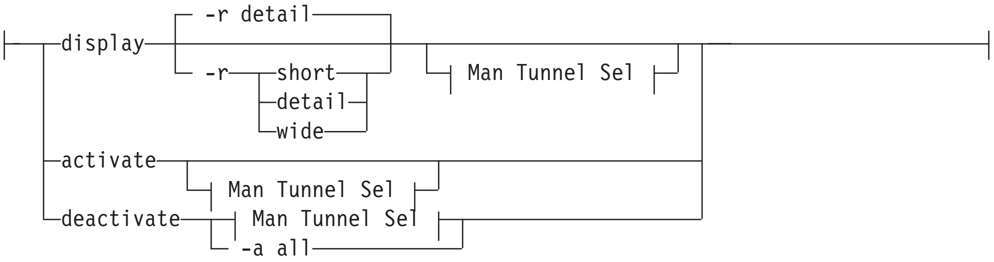
IP Filter Option:



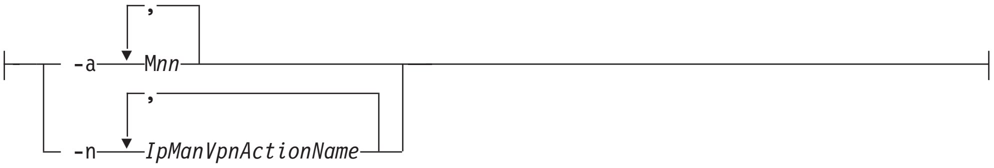
Filter Selection:



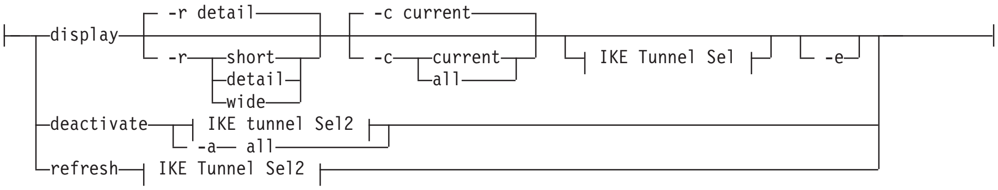
Manual Tunnel Option:



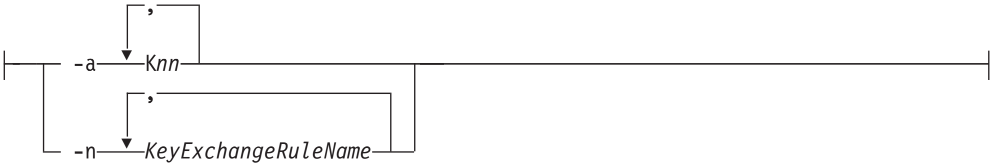
Man Tunnel Selection:



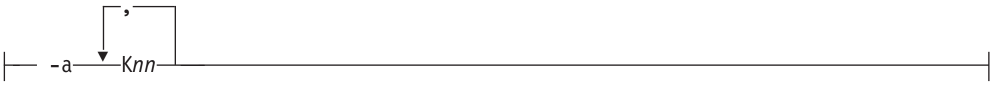
IKE Tunnel Option:



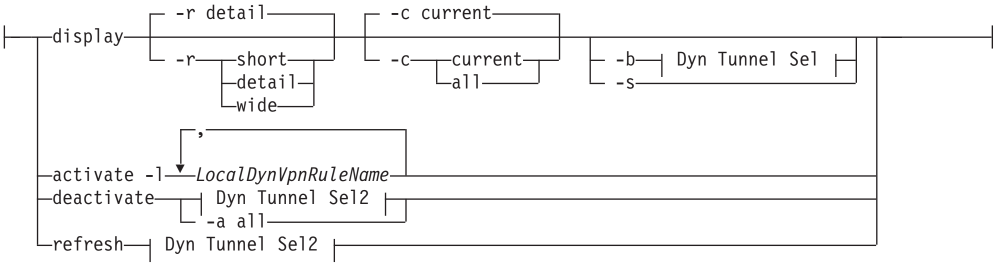
IKE Tunnel Selection:



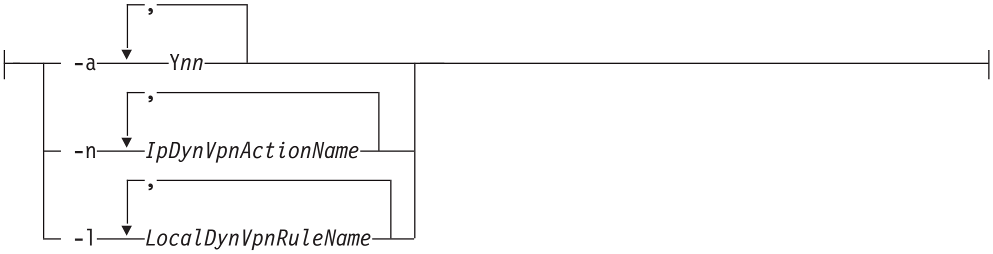
IKE Tunnel Selection2:



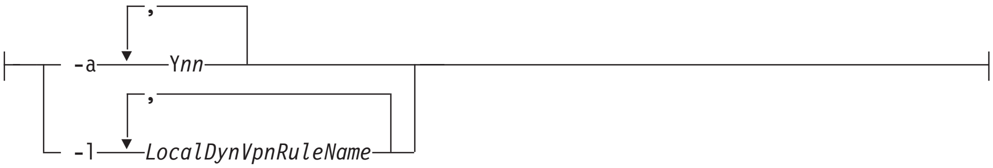
Dynamic Tunnel Option:



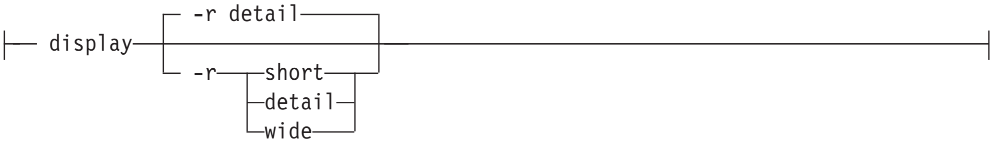
Dyn Tunnel Selection:



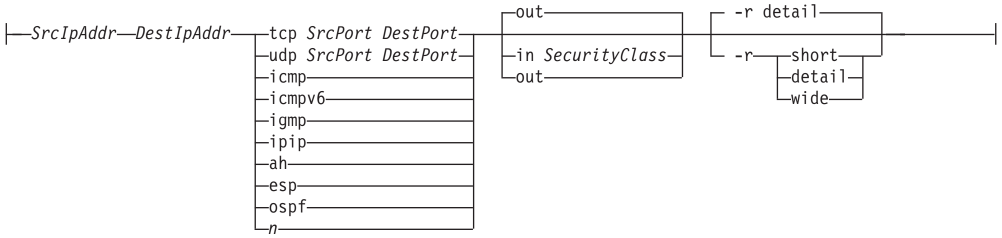
Dyn Tunnel Selection2:



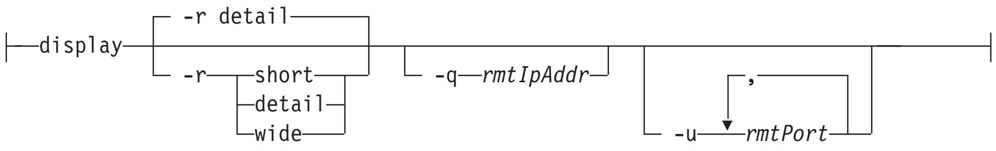
Interface Option:



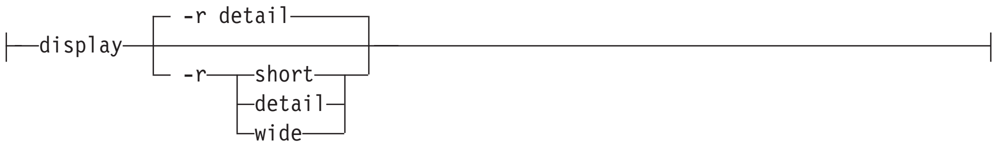
IP Traffic Test Option:



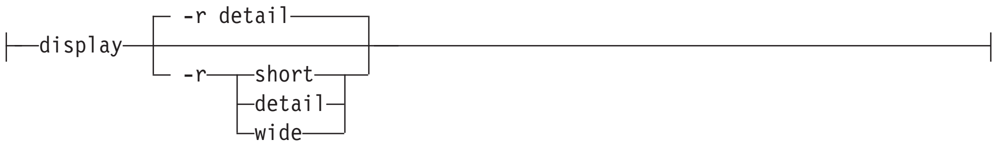
NATT Port Translation Option:



IKED Network Security Option:



Network Security Server Option:



The z/OS UNIX ipsec command parameter descriptions

The following topics describe the individual parameter items that are identified in the syntax diagram. 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** could be specified as **-f DEF**. Option values for -n, -g, and -l specify a name and are case sensitive. Option values for -p and -z specify a name and are not case sensitive.

Primary options

-f Display or modify IP 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.

Global options

-d

Generates debug information during command execution. An optional *debuglevel* value can be specified along with the -d option. Debug output is sent to stderr or to stdout, according to 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 selected criteria that was specified when the command was issued.

The selective level debug can be specified with any option. Selective debug information is available only for -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.

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.

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 security information in a given format. The default format is `detail`. See "General report concepts" on page 554 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 in use by the stack. Filter rules that are inactive because of time conditions are not 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.

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.

-a M*nn* **or -a Y***nn*

Displays the IP 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 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 to the rule. The integer corresponds to the order in which the 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 matching the *IpFilterRuleName* value are displayed.

-g *IpFilterGroupName*

Specifies one or more IP filter groups 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 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 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.

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 filter rules as 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 can resume, including refreshes and new activations. If the IKE daemon is active, it will attempt to perform all configured autoactivations.

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 rules originate as default profile rules or as policy rules.

See also “IP filter (-f) primary option” on page 560 for report details and examples.

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 M***xx* 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 “General report concepts” on page 554 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 570 for report details and examples.

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 “General report concepts” on page 554 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 superceded 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: All dynamic tunnels that are associated with deactivated IKE tunnels will also be deactivated as part of this request.

Restriction: You should 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. Configuration options typically cause a refresh (and therefore a new set of cryptographic keys) on a lifetime or lifesize basis.

Restriction: You should 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 573 for report details and examples.

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 "General report concepts" on page 554 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 in the *z/OS Communications Server: IP Configuration Guide* for more information.

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).

Restriction: You should 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: You should 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 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 577 for report details and examples.

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 “General report concepts” on page 554 for a description of the different report formats.

See also “Interface (-i) primary option” on page 591 for report details and examples.

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. Protocol number 0 indicates to match any protocol.

SrcPort
DestPort

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 1 024 - 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 a value of 0 should be specified on input to the traffic test. For more information about NAT traversal and remote port translation, see 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 “General report concepts” on page 554 for a description of the different report formats.

See also “IP traffic test (-t) primary option” on page 592 for report details and examples.

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 “General report concepts” on page 554 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 597 for report details and examples.

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 "General report concepts" for a description of the different report formats.

Network security server (-x) option

Parameters

display
Display information for each NSS 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 client that is connected to the server.

-r *format*
Display network security information in a given format. The default format is detail. See "General report concepts" for a description of the different report formats.

The ipsec command report details and examples

This sections contains descriptive information on the formatting and contents of **ipsec** reports, including examples.

General report concepts

In order to fully understand the following concepts and fields, you should have some general knowledge of IP Security. See IP security in the *z/OS Communications Server: IP Configuration Guide* for more information.

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 is an example of a short format.

ipsec -f display -r short

```
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 14:52:53 2006
Primary:  Filter          Function: Display            Format:   Short
Source:   Stack Policy    Scope:    Current            TotAvail: 144
Logging:  On              Predecap: Off                DVIPSec:  Yes
NatKeepAlive:  20

FilterName  |FilterNameExtension
            |GroupName
            |LocalStartActionName
            |VpnActionName
            |TunnelID
            |Type|State|Action|Scope|Direction|OnDemand
            |SecurityClass|Logging
            |Protocol|ICMPType|ICMPCode|OSPFType|TCPQualifier|ProtocolGran
            |SourceAddress
            |SourceAddressPrefix
            |SourceAddressRange
            |SourceAddressGran
            |SourcePort|SourcePortRange|SourcePortGran
            |DestAddress
            |DestAddressPrefix
            |DestAddressRange
            |DestAddressGran
            |DestPort|DestPortRange|DestPortGran
            |OrigRmtConnPort|RmtIDpayload|RmtUdpEncapPort
            |CreateTime|UpdateTime
IPSecGWv4 |1
            |n/a
            |IPSecGWv4_start
            |IPSec__Gold
            |Y9
            |Dynamic|Active|Permit|Routed|Outbound|Yes
            |0|All
            |All|n/a|n/a|n/a|n/a|Rule
            |10.81.2.1
            |n/a
            |n/a
            |Packet
            |All|n/a|Rule
            |10.81.8.1
            |n/a
            |n/a
            |Packet
            |All|n/a|Rule
            |n/a|n/a|n/a
            |n/a|n/a
......
IPSecGWv6 |1
            |n/a
            |IPSecGWv6_start
            |IPSec__Gold
            |Y11
            |Dynamic|Active|Permit|Routed|Outbound|Yes
```

```
|0|All
|All|n/a|n/a|n/a|n/a|Rule
|2001:db8:10::81:2:1
|n/a
|n/a
|Packet
|All|n/a|Rule
|2001:db8:10::81:8:1
|n/a
|n/a
|Packet
|All|n/a|Rule
|n/a|n/a|n/a
|n/a|n/a
```

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 -f display -r detail
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 14:57:48 2006
Primary:  Filter          Function: Display            Format:   Detail
Source:   Stack Policy    Scope:    Current            TotAvail: 144
Logging:  On              Predecap: Off                DVIPSec:  Yes
NatKeepAlive:  20

FilterName:                   IPSecGWv4
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         IPSecGWv4_start
VpnActionName:                IPSec__Gold
TunnelID:                     Y9
Type:                         Dynamic
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                10.81.2.1
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.81.8.1
DestAddressPrefix:            n/a
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
**********************************************************************
......
FilterName:                 IPSecGWv6
FilterNameExtension:        1
GroupName:                  n/a
LocalStartActionName:       IPSecGWv6_start
VpnActionName:              IPSec__Gold
TunnelID:                   Y11
Type:                       Dynamic
State:                      Active
Action:                     Permit
Scope:                      Routed
Direction:                  Outbound
OnDemand:                   Yes
SecurityClass:              0
Logging:                    All
Protocol:                   All
ICMPType:                   n/a
ICMPCode:                   n/a
OSPFType:                   n/a
TCPQualifier:               n/a
ProtocolGranularity:        Rule
SourceAddress:              2001:db8:10::81:2:1
SourceAddressPrefix:        n/a
SourceAddressRange:         n/a
SourceAddressGranularity:   Packet
SourcePort:                 All
SourcePortRange:            n/a
SourcePortGranularity:      Rule
DestAddress:                2001:db8:10::81:8:1
DestAddressPrefix:          n/a
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
**********************************************************************
```

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 -f display -r wide
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 15:01:00 2006
Primary:  Filter         Function: Display           Format:   Wide
Source:   Stack Policy   Scope:    Current           TotAvail: 144
Logging:  On             Predecap: Off               DVIPSec:  Yes
NatKeepAlive:  20

FilterName|FilterNameExtension|GroupName|LocalStartActionName|VpnActionName|Tunn
elID|Type|State|Action|Scope|Direction|OnDemand|SecurityClass|Logging|Protocol|I
CMPType|ICMPCode|OSPFType|TCPQualifier|ProtocolGran|SourceAddress|SourceAddressP
refix|SourceAddressRange|SourceAddressGran|SourcePort|SourcePortRange|SourcePort
```

```
Gran|DestAddress|DestAddressPrefix|DestAddressRange|DestAddressGran|DestPort|Des
tPortRange|DestPortGran|OrigRmtConnPort|RmtIDPayload|RmtUdpEncapPort|CreateTime|
UpdateTime
IPSecGWv4|1|n/a|IPSecGWv4_start|IPSec__Gold|Y9|Dynamic|Active|Permit|Routed|Outb
ound|Yes|0|All|All|n/a|n/a|n/a|n/a|Rule|10.81.2.1|n/a|n/a|Packet|All|n/a|Rule|10
.81.8.1|n/a|n/a|Packet|All|n/a|Rule|n/a|n/a|n/a|n/a|n/a
......
IPSecGWv6|1|n/a|IPSecGWv6_start|IPSec__Gold|Y0|Dynamic Anchor|Active|Permit|Rout
ed|Outbound|Yes|0|All|All|n/a|n/a|n/a|n/a|Rule|2001:db8:10::81:2:1|n/a|n/a|Packe
t|All|n/a|Rule|2001:db8:10::81:8:1|n/a|n/a|Packet|All|n/a|Rule|n/a|n/a|n/a|2006/
01/24 08:12:57|2006/01/24 08:45:32
```

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 fourth and possibly fifth 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:

```
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 14:52:53 2006
Primary:  Filter         Function: Display           Format:   Short
Source:   Stack Policy   Scope:    Current           TotAvail: 144
Logging:  On             Predecap: Off               DVIPSec:  Yes
NatKeepAlive:  20
```

The first heading line shows the following fields:

Stack Name
: The stack name with which the command is associated.

NSS Client Name
: The name 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`, `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 shows as `display (cascade)`. If the request is for shadow dynamic tunnels (**-y dis -s**), then the function field shows as `display (shadows)`.

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

For the Filter (-f) primary option, the source is one of the following:

- Stack Profile - Data is from the default IP filter policy specified in the IP stack's profile
- Stack Policy - Data is from the IP filter policy specified by the Policy Agent

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 560 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) 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 (defaulting **Scope** to `current`) might format three tunnel entries, while **TotAvail** indicates a value of 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) primary option, the fourth heading line shows:

Logging

Indicates (at a global level), whether packet filter logging is in use.

- If the value of Source is Stack Profile, that value reflects the setting of the LOGENABLE/LOGDISABLE keyword of the IPSEC statement.
- If the value of Source is Stack Policy, that value reflects the FilterLogging setting of the IpFilterPolicy statement.

Predecap

Indicates whether decapsulated packets are first filtered at the stack.

- If the value of Source is Stack Profile, that value is equal to No.
- If the value of Source is Stack Policy, that value reflects the PreDecap setting on the IpFilterPolicy statement.

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 reflects the setting of the DVIPSEC keyword of the IPSEC statement in the TCP/IP profile. It applies to the treatment of both Stack Profile and Stack Policy filters.

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) primary option, 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.

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.

Report data

Independent of the output format, all data fields are shown, even if some of the fields are not applicable to the type of entry that is being displayed or 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

Purpose

The -f primary option is used to display and manage IP filter rules that are used in the TCP/IP stack. All IP Security traffic is ultimately controlled through the use of the filter rules that are current in the stack. The current filter rules can come 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).

See “IP filter (-f) option” on page 547 for parameter descriptions.

Syntax

See “The z/OS UNIX ipsec command syntax” on page 541 for -f primary option syntax.

Command examples

ipsec -f display -c profile
Displays the profile 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 filter rule set that was obtained through the Policy Agent to the default IP filter policy that was specified in the TCPIP profile. The request is directed to the NSS server.

ipsec -f reload
Changes the IP filter rule set from the default IP filter policy that was specified in the TCPIP profile to the IP filter policy that was created in the Policy Agent.

Report example

`ipsec -f dis -p tcpcs1 -a y3`

```
CS V1R9 ipsec  Stack Name: TCPCS1    Mon Jun 14 14:48:50 2004
Primary:  Filter          Function: display           Format:   detail
Source:   Stack Policy    Scope:    current           TotAvail: 32
Logging:  Yes             Predecap: Yes               DVIPSec:  Yes
NatKeepAlive:  20

FilterName:                   TCPCS1-SWSA
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                E-N-TRAN-AHMD5-DES
TunnelID:                     Y3
Type:                         Dynamic
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     YES
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                10.0.0.1
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.22.22.1
DestAddressPrefix:            n/a
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
*************************************************************************
FilterName:                   TCPCS1-SWSA
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                E-N-TRAN-AHMD5-DES
TunnelID:                     Y0
Type:                         Dynamic Anchor
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     YES
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                10.0.0.1
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  10.22.22.0
DestAddressPrefix:            24
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2004/06/14 08:12:57
UpdateTime:                   2004/06/14 08:12:57
*************************************************************************

2 entries selected
ipsec -f dis -p tcpcs1 -a y2 -h

CS V1R9 ipsec  Stack Name: TCPCS1    Mon Jun 14 14:48:50 2004
Primary:  Filter          Function: display            Format:   detail
Source:   Stack Policy    Scope:    current            TotAvail: 40
Logging:  Yes             Predecap: Yes                DVIPSec:  Yes
NatKeepAlive:  20

FilterName:                   LocalNtt_Log
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         LocalStartActNtt
VpnActionName:                DynVpnAct
TunnelID:                     Y2
Type:                         NATT Dynamic
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     No
SecurityClass:                0
Logging:                      All
Protocol:                     TCP(6)
```

```
    ICMPType:                     n/a
    ICMPCode:                     n/a
    OSPFType:                     n/a
    TCPQualifier:                 None
    ProtocolGranularity:          Rule
    SourceAddress:                9.42.105.144
    SourceAddressPrefix:          n/a
    SourceAddressRange:           n/a
    SourceAddressGranularity:     Packet
    SourcePort:                   All
    SourcePortRange:              n/a
    SourcePortGranularity:        Rule
    DestAddress:                  9.42.105.3
    DestAddressPrefix:            n/a
    DestAddressRange:             n/a
    DestAddressGranularity:       Packet
    DestPort:                     All
    DestPortRange:                n/a
    DestPortGranularity:          Rule
    OrigRmtConnPort:              n/a
    RmtIDPayload:                 192.168.90.1
    RmtUdpEncapPort:              4500
|   CreateTime:                   n/a
|   UpdateTime:                   n/a
|   ************************************************************************
    FilterName:                   LocalNtt_Log
    FilterNameExtension:          1
    GroupName:                    n/a
    LocalStartActionName:         LocalStartActNtt
    VpnActionName:                DynVpnAct
    TunnelID:                     Y2
    Type:                         NRF
    State:                        Active
    Action:                       Permit
    Scope:                        Local
    Direction:                    Outbound
    OnDemand:                     No
    SecurityClass:                0
    Logging:                      All
    Protocol:                     TCP(6)
    ICMPType:                     n/a
    ICMPCode:                     n/a
    OSPFType:                     n/a
    TCPQualifier:                 None
    ProtocolGranularity:          Rule
    SourceAddress:                9.42.105.144
    SourceAddressPrefix:          n/a
    SourceAddressRange:           n/a
    SourceAddressGranularity:     Packet
    SourcePort:                   3004
    SourcePortRange:              n/a
    SourcePortGranularity:        Rule
    DestAddress:                  9.42.105.3
    DestAddressPrefix:            n/a
    DestAddressRange:             n/a
    DestAddressGranularity:       Packet
    DestPort:                     3005
    DestPortRange:                n/a
    DestPortGranularity:          Rule
    OrigRmtConnPort:              3005
    RmtIDPayload:                 n/a
    RmtUdpEncapPort:              n/a
|   CreateTime:                   n/a
|   UpdateTime:                   n/a
|   ************************************************************************
    FilterName:                   LocalNtt_Log
    FilterNameExtension:          1
```

```
GroupName:                    n/a
LocalStartActionName:         LocalStartActNtt
VpnActionName:                DynVpnAct
TunnelID:                     Y0
Type:                         NATT Anchor
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     No
SecurityClass:                0
Logging:                      All
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                9.42.105.144
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  9.42.105.3
DestAddressPrefix:            n/a
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
***********************************************************************
FilterName:                   LocalNtt_Log
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         LocalStartActNtt
VpnActionName:                DynVpnAct
TunnelID:                     Y0
Type:                         Dynamic Anchor
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     No
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                9.42.105.0
SourceAddressPrefix:          24
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  9.42.105.3
DestAddressPrefix:            n/a
```

```
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
CreateTime:                   2004/06/14 08:12:57
UpdateTime:                   2004/06/14 08:12:57
*********************************************************************

4 entries selected
```

Note: The inbound entries were truncated from the previous example. They have the same information format as the outbound entries displayed in that example.

Report field descriptions

For more information about the header, see “Report heading” on page 558.

FilterName
All filter rules have a base name that is used for reference purposes. This base FilterName value is assigned by the system for filters that were created from 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 specifying an **ipsec** command selection criteria using -n.

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 a value of `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:

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

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` or `Deny`.

| **Result:** 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`.

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. A value of 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.

Logging
Indicates the logging 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), or None (no log entries are generated from this filter).

Protocol
Indicates the protocol to which the filter applies. A value of 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, or 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 shows the type of ICMP message as specified. If no ICMP type is specified, the value is All. If the protocol is not ICMP, this field is not applicable.

ICMPCode
If the protocol is ICMP or ICMPv6, this field shows the code of ICMP message as specified. If no ICMP code is specified, the value is All. If the protocol is not ICMP, this field is not applicable.

OSPFType
If the protocol is OSPF, this field shows the type of OSPF as specified. If no OSPF type is specified, the value is All. If the protocol is not OSPF, this field is not applicable.

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. A value of Rule indicates that dynamic tunnel negotiation uses the protocol specification from the matching filter rule. A value of Packet indicates that the dynamic tunnel negotiation uses the protocol specification from the packet that initiated the dynamic tunnel activation.

SourceAddress
The source IP address to which this filter applies.

- If the SourceAddressPrefix and SourceAddressRange fields both indicate a value of `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 is 0 or the SourceAddressRange 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, a SourceAddressPrefix value of 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 of 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. A value of `Rule` indicates that dynamic tunnel negotiation will use the source IP address specification from the matching filter rule. A value of `Packet` indicates that dynamic tunnel negotiation will use the source IP address specification from the packet that initiated the dynamic tunnel activation.

SourcePort

This field contains the source port to which this filter applies. If a port range is specified, then SourcePort is the first port number of the range. If the value is `0`, the filter applies to all source ports.

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

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 source port specification from the matching filter rule. A value of `Packet` indicates that dynamic tunnel negotiation uses the source port specification from the packet that initiated the dynamic tunnel activation.

DestAddress

The destination IP address to which this filter applies.

- If the DestAddressPrefix and DestAddressRange fields both indicate a value of 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 source 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 has a value of 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. A value of Rule indicates that dynamic tunnel negotiation will use the destination IP address specification from the matching filter rule. A value of Packet indicates that dynamic tunnel negotiation will use the destination IP address specification from the packet that initiated the dynamic tunnel activation.

DestPort

Contains 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 value is 0, the filter applies to all destination ports.

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

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.

OrigRmtConnPort
The remote endpoint's original connection port value. This field is applicable only for NRF filter entries. When the value is nonzero, remote port translation is being done and a connection's remote port value 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. 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 filter's attributes 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`.

Manual tunnel (-m) primary option

Purpose

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 "Manual tunnel (-m) option" on page 549 for parameter descriptions.

Syntax

For -m primary option syntax see "The z/OS UNIX ipsec command syntax" on page 541.

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.

Report example

`ipsec -m display`

```
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 15:41:20 2006
Primary:  Manual tunnel   Function: Display           Format:   Detail
Source:   Stack           Scope:    Current            TotAvail: 2

TunnelID:                     M1
VpnActionName:                Manual-Tunnel-IPv6
State:                        Active
HowToEncap:                   Transport
LocalEndPoint:                2001:db8:10::51:0:4
RemoteEndPoint:               2001:db8:10::51:0:5
HowToAuth:                    ESP
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              1902
 AuthOutboundSpi:             5119
HowToEncrypt:                 AES
 EncryptInboundSpi:           1902
 EncryptOutboundSpi:          5119
OutboundPackets:              0
OutboundBytes:                0
InboundPackets:               0
InboundBytes:                 0
***********************************************************************
TunnelID:                     M2
VpnActionName:                Manual-Tunnel-IPv4
State:                        Inactive
HowToEncap:                   Transport
LocalEndPoint:                10.51.0.4
RemoteEndPoint:               10.51.0.5
HowToAuth:                    ESP
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              3755
 AuthOutboundSpi:             4081
HowToEncrypt:                 AES
 EncryptInboundSpi:           3755
 EncryptOutboundSpi:          4081
OutboundPackets:              0
OutboundBytes:                0
InboundPackets:               0
InboundBytes:                 0
***********************************************************************

2 entries selected
```

Report field descriptions

For more information about the header, see “Report heading” on page 558.

TunnelID
: The ID that uniquely defines the manual tunnel. In this example, TunnelID has a value of `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 a value of `Active` or `Inactive`. `Active` indicates that the tunnel is available for use between the local endpoint and remote endpoint. `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 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 `Hmac_Md5` or `Hmac_Sha`.

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, what encryption algorithm will be used. Possible values are `DES`, `3DES`, `or` `AES`.

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.

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.

IKE tunnel (-k) primary option

Purpose

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 "IKE tunnel (-k) option" on page 549 for parameter descriptions.

Syntax

For -k primary option syntax see "The z/OS UNIX ipsec command syntax" on page 541.

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). 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.

Report example

ipsec -k display

```
CS V1R9 ipsec  Stack Name: TCPCS4  Tue Jan 24 15:46:51 2006
Primary:  IKE tunnel      Function: Display           Format:   Detail
Source:   IKED            Scope:    Current           TotAvail: n/a

TunnelID:                     K1
KeyExchangeRuleName:          IPSecGWv4
KeyExchangeActionName:        Aggr-SHA-AES
LocalEndPoint:                10.11.5.4
LocalIDType:                  X500DN
LocalID:                      CN=MVS172_IKE,OU=FVT,O=IBM,L=RTP,ST=NC,C=US
RemoteEndPoint:               10.11.4.5
RemoteIDType:                 X500DN
RemoteID:                     CN=MVS202_IKE,OU=FVT,O=IBM,L=RTP,ST=NC,C=US
ExchangeMode:                 Aggressive
State:                        DONE
```

```
AuthenticationAlgorithm:      Hmac_Sha
EncryptionAlgorithm:          AES
DiffieHellmanGroup            5
AuthenticationMethod:         RSASignature
InitiatorCookie:              0XF9F3962437723A4F
ResponderCookie:              0XA75BBF99E7A3BF3B
Lifesize:                     0K
CurrentByteCount:             960b
Lifetime:                     480m
LifetimeRefresh:              2006/01/24 20:33:22
LifetimeExpires:              2006/01/24 22:44:39
Role:                         Initiator
AssociatedDynamicTunnels:     2
NATTSupportLevel:             None
NATInFrntLclScEndPnt:         No
NATInFrntRmtScEndPnt:         No
zOSCanInitiateP1SA:           Yes
AllowNat:                     No
RmtNAPTDetected:              No
RmtUdpEncapPort:              n/a
***********************************************************************
TunnelID:                     K4
KeyExchangeRuleName:          IPSecGWv6
KeyExchangeActionName:        Aggr-SHA-AES
LocalEndPoint:                2001:db8:10::12:5:4
LocalIDType:                  X500DN
LocalID:                      CN=MVS172_IKE,OU=FVT,O=IBM,L=RTP,ST=NC,C=US
RemoteEndPoint:               2001:db8:10::12:4:5
RemoteIDType:                 X500DN
RemoteID:                     CN=MVS202_IKE,OU=FVT,O=IBM,L=RTP,ST=NC,C=US
ExchangeMode:                 Aggressive
State:                        DONE
AuthenticationAlgorithm:      Hmac_Sha
EncryptionAlgorithm:          AES
DiffieHellmanGroup            5
AuthenticationMethod:         RSASignature
InitiatorCookie:              0X2FEC7085C8F3B255
ResponderCookie:              0X34469E07CC7C1F7B
Lifesize:                     0K
CurrentByteCount:             1088b
Lifetime:                     480m
LifetimeRefresh:              2006/01/24 21:27:14
LifetimeExpires:              2006/01/24 22:44:39
Role:                         Initiator
AssociatedDynamicTunnels:     2
NATTSupportLevel:             n/a
NATInFrntLclScEndPnt:         n/a
NATInFrntRmtScEndPnt:         n/a
zOSCanInitiateP1SA:           n/a
AllowNat:                     n/a
RmtNAPTDetected:              n/a
RmtUdpEncapPort:              n/a
***********************************************************************

2 entries selected
```

Report field descriptions

For more information about the header, see “Report heading” on page 558.

TunnelID

The ID that uniquely defines the IKE tunnel. In this example, TunnelID has a value of 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.

KeyExchangeRuleName
The name of the KeyExchangeRule statement that was used to define and control the characteristics of the IKE tunnel.

KeyExchangeActionName
The name of the KeyExchangeAction statement that was used to initiate the IKE tunnel.

LocalEndpoint
The local security endpoint address of the IKE tunnel.

LocalIDType
Specifies the type of the local identity. Possible values are:

IPV4 An IPv4 address.

IPV6 An IPv6 address.

FQDN
A fully qualified domain name.

USERFQDN
A user at a fully qualified domain name.

X500DN
An X.500 distinguished name.

LocalID
Specifies the value of the local identity.

RemoteEndpoint
The remote security endpoint address of the IKE tunnel.

RemoteIDType
Specifies the type of the remote identity. Possible values are:

IPV4 An IPv4 address.

IPV6 An IPv6 address.

FQDN
A fully qualified domain name.

USERFQDN
A user at a fully qualified domain name.

X500DN
An X.500 distinguished name.

RemoteID
Specifies the value of the remote identity.

ExchangeMode
The exchange mode used to negotiate the IKE tunnel. Possible values are Aggressive or Main.

State 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.

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 on 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.

AuthenticationAlgorithm
Specifies the authentication algorithm that is used for authenticating IKE key exchange messages. Possible values are `HMAC_MD5` or `HMAC_SHA`.

EncryptionAlgorithm
Specifies the encryption algorithm that is used for protecting IKE key exchange messages. Possible values are `DES`, `3DES`, or `AES`.

DiffieHellmanGroup
Indicates the DiffieHellmanGroup value that is used during key exchange. If no DiffieHellmanGroup is used, the value is `0`.

AuthenticationMethod
Indicates the method of authenticating the peer endpoint. Possible values are `PresharedKey` or `RsaSignature`.

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 negotiated 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 negotiated number of minutes between each refresh.

LifetimeRefresh
The time at which the tunnel must be refreshed.

LifetimeExpires
The time at which the tunnel expires.

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 following are possible values:

D2RFC
Draft 2 of RFC.

D3RFC
Draft 3 of RFC.

RFC RFC, with a non-z/OS remote security endpoint.

RFC_zOS
RFC, with a z/OS remote security endpoint.

n/a NAT traversal is not supported for phase 1 SAs using 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 or not z/OS can initiate the initial phase 1 SA negotiation. NAT traversal is not supported for phase 1 SAs using IPv6 addresses. In this case, the field has the value `n/a`.

AllowNat
Indicates whether NAT traversal support is enabled. This field reflects the configured setting of the AllowNat keyword. NAT traversal is not supported for phase 1 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. 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 using 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 a value of n/a.

Dynamic tunnel (-y) primary option

Purpose

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 "Dynamic tunnel (-y) option" on page 551 for parameter descriptions.

Syntax

For -y primary option syntax see "The z/OS UNIX ipsec command syntax" on page 541.

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.

Report examples

```
ipsec -y display
CS V1R9 ipsec  Stack Name: TCPCS4  Fri Feb  3 12:02:08 2006
Primary:  Dynamic tunnel  Function: Display           Format:   Detail
Source:   Stack           Scope:    Current           TotAvail: 26
......
TunnelID:                     Y87
ParentIKETunnelID:            K82
VpnActionName:                Hmac-SHA_AES
LocalDynVpnRule:              n/a
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                10.81.4.4
RemoteEndPoint:               10.81.5.5
LocalAddressBase:             0.0.0.0
LocalAddressPrefix:           0
LocalAddressRange:            n/a
RemoteAddressBase:            10.81.8.0
RemoteAddressPrefix:          24
RemoteAddressRange:           n/a
HowToAuth:                    ESP
```

```
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              3736726637
 AuthOutboundSpi:             3849751685
HowToEncrypt:                 AES
 EncryptInboundSpi:           3736726637
 EncryptOutboundSpi:          3849751685
Protocol:                     ICMP(1)
LocalPort:                    0
RemotePort:                   0
OutboundPackets:              1
OutboundBytes:                284
InboundPackets:               1
InboundBytes:                 284
Lifesize:                     200K
LifesizeRefresh:              168K
CurrentByteCount:             568b
LifetimeRefresh:              2006/02/03 12:17:15
LifetimeExpires:              2006/02/03 12:21:55
CurrentTime:                  2006/02/03 12:02:08
VPNLifeExpires:               2006/04/29 20:51:55
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
***********************************************************************
......
TunnelID:                     Y129
ParentIKETunnelID:            K124
VpnActionName:                Hmac-SHA_AES
LocalDynVpnRule:              n/a
State:                        Active
HowToEncap:                   Tunnel
LocalEndPoint:                2001:db8:10::81:4:4
RemoteEndPoint:               2001:db8:10::81:5:5
LocalAddressBase:             ::
LocalAddressPrefix:           0
LocalAddressRange:            n/a
RemoteAddressBase:            2001:db8:10::81:8:0
RemoteAddressPrefix:          112
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              4260457118
 AuthOutboundSpi:             2031095091
HowToEncrypt:                 AES
 EncryptInboundSpi:           4260457118
 EncryptOutboundSpi:          2031095091
Protocol:                     ICMPV6(58)
LocalPort:                    0
RemotePort:                   0
OutboundPackets:              1
OutboundBytes:                304
InboundPackets:               1
InboundBytes:                 344
Lifesize:                     200K
LifesizeRefresh:              158K
CurrentByteCount:             648b
LifetimeRefresh:              2006/02/03 12:19:12
LifetimeExpires:              2006/02/03 12:25:28
CurrentTime:                  2006/02/03 12:03:02
```

```
VPNLifeExpires:              2006/04/29 20:55:28
NAT Traversal Topology:
  UdpEncapMode:              n/a
  LclNATDetected:            n/a
  RmtNATDetected:            n/a
  RmtNAPTDetected:           n/a
  RmtIsGw:                   n/a
  RmtIsZOS:                  n/a
  zOSCanInitP2SA:            n/a
  RmtUdpEncapPort:           n/a
  SrcNATOARcvd:              n/a
  DstNATOARcvd:              n/a
***********************************************************************
......
22 entries selected
```

Report field descriptions

For more information about the header, see “Report heading” on page 558.

TunnelID
The ID that uniquely defines the dynamic tunnel. In this example, TunnelID has a value of `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. If the dynamic tunnel entry is a shadow tunnel from a FIREWALL-configured distributor, the TunnelID is `n/a`.

ParentIKETunnelID
The tunnel ID of the phase 1 (IKE) tunnel that enables the creation of this dynamic tunnel.

FirewallTunnelName
If the dynamic tunnel entry is from a FIREWALL-configured distributor, then the FirewallTunnelName field is displayed rather than the VpnActionName field. This is a 76-byte field carrying the raw tunnel name as defined by the Firewall Technologies product.

Rule: This field is displayed only with a request for shadow tunnels.

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`.

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 `Hmac_Md5` or `Hmac_Sha`.

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 `DES`, `3DES`, `AES`, or `NULL`.

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.

Protocol
Indicates the protocol that the dynamic tunnel is protecting. A value of 0 indicates that the dynamic tunnel is protecting all protocols. This field is not applicable for IPv6 OSPF manual tunnels.

LocalPort
Indicates the source port number of the data traffic that the dynamic tunnel is protecting. A value of 0 indicates that the dynamic tunnel is protecting all source ports. This field is not applicable for IPv6 OSPF manual tunnels.

RemotePort
Indicates the destination port number of the data traffic that the dynamic tunnel is protecting. A value of 0 indicates that the dynamic tunnel is protecting all destination ports. This field is not applicable for IPv6 OSPF manual tunnels.

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 can be protected by the tunnel before it must be refreshed. If the value is 0, then the negotiated refresh lifesize was `None` and byte counts are not used to monitor for tunnel refresh.

LifesizeRefresh

The number of kilobytes that can 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 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 negotiated 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 negotiated 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.

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: IP address in the source NAT-OA payload 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-OA payload was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a source NAT-OA payload. For information about accessing RFCs, see Appendix F, "Related protocol specifications," on page 891.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

DstNATOARcvd
For a UDP-encapsulated transport SA: IP address in the destination NAT-OA payload 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 host's public address. This value is 0.0.0.0 if a destination NAT-OA payload was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a destination NAT-OA payload. For information about accessing RFCs, see Appendix F, "Related protocol specifications," on page 891.

For a non-UDP-encapsulated transport SA: The value is `n/a`.

Report examples

`ipsec -y display -b`

```
CS V1R9 ipsec  Stack Name: TCPCS4  Fri Feb  3 12:10:42 2006
Primary:  Dynamic tunnel  Function: Display           Format:   Detail
Source:   IKED            Scope:    Current           TotAvail: n/a

TunnelID:                     Y87
ParentIKETunnelID:            K82
VpnActionName:                Hmac-SHA_AES
LocalDynVpnRule:              n/a
IpFilterRule:                 n/a
```

```
State:                        DONE
HowActivated:                 OnDemand
HowToEncap:                   Tunnel
LocalEndPoint:                10.81.4.4
RemoteEndPoint:               10.81.5.5
LocalAddressBase:             0.0.0.0
LocalAddressPrefix:           0
LocalAddressRange:            n/a
RemoteAddressBase:            10.81.8.0
RemoteAddressPrefix:          24
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              3736726637
 AuthOutboundSpi:             3849751685
HowToEncrypt:                 AES
 EncryptInboundSpi:           3736726637
 EncryptOutboundSpi:          3849751685
Lifesize:                     200K
LifesizeRefresh:              168K
LifetimeRefresh:              2006/02/03 12:17:15
LifetimeExpires:              2006/02/03 12:21:55
CurrentTime:                  2006/02/03 12:10:42
VPNLifeExpires:               2006/04/29 20:51:55
AssociatedFiltProtocol:       ICMP(1)
AssociatedFiltSrcPort:        0
AssociatedFiltDestPort:       0
PFS                           Yes
DiffieHellmanGroup:           14
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:                     Y129
ParentIKETunnelID:            K124
VpnActionName:                Hmac-SHA_AES
LocalDynVpnRule:              n/a
IpFilterRule:                 n/a
State:                        DONE
HowActivated:                 OnDemand
HowToEncap:                   Tunnel
LocalEndPoint:                2001:db8:10::81:4:4
RemoteEndPoint:               2001:db8:10::81:5:5
LocalAddressBase:             ::
LocalAddressPrefix:           0
LocalAddressRange:            n/a
RemoteAddressBase:            2001:db8:10::81:8:0
RemoteAddressPrefix:          112
RemoteAddressRange:           n/a
HowToAuth:                    ESP
 AuthAlgorithm:               Hmac_Sha
 AuthInboundSpi:              4260457118
 AuthOutboundSpi:             2031095091
HowToEncrypt:                 AES
 EncryptInboundSpi:           4260457118
 EncryptOutboundSpi:         2031095091
Lifesize:                    200K
LifesizeRefresh:             158K
LifetimeRefresh:             2006/02/03 12:19:12
LifetimeExpires:             2006/02/03 12:25:28
CurrentTime:                 2006/02/03 12:10:47
VPNLifeExpires:              2006/04/29 20:55:28
AssociatedFiltProtocol:      ICMPV6(58)
AssociatedFiltSrcPort:       0
AssociatedFiltDestPort:      0
PFS                          Yes
DiffieHellmanGroup:          14
PendingNewActivation:        n/a
NAT Traversal Topology:
  UdpEncapMode:              n/a
  LclNATDetected:            n/a
  RmtNATDetected:            n/a
  RmtNAPTDetected:           n/a
  RmtIsGw:                   n/a
  RmtIsZOS:                  n/a
  zOSCanInitP2SA:            n/a
  RmtUdpEncapPort:           n/a
  SrcNATOARcvd:              n/a
  DstNATOARcvd:              n/a
  LclIpSpecExIDPayload:      n/a
  RmtIpSpecExIDPayload:      n/a
************************************************************************
......
22 entries selected
```

Report field descriptions

For more information about the header, see “Report heading” on page 558.

TunnelID
The ID that uniquely defines the dynamic tunnel. In this example, the TunnelID has a value of `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`.

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`.

IpFilterRule
The name of the dynamic anchor rule in POLICY that controlled the creation of the dynamic tunnel.

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 Hmac_Md5 or Hmac_Sha.

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 DES, 3DES, AES, NULL, or n/a.

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.

Lifesize

The number of kilobytes that can can be protected by the tunnel before it must be refreshed. If the value is 0, then the negotiated refresh lifesize value was None; byte counts are not used to monitor for tunnel refresh.

LifesizeRefresh

The number of kilobytes that can 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. A value 0 indicates that the dynamic tunnel is protecting all protocols.

AssociatedFiltSrcPort
Indicates the source port number of the data traffic that is being protected by the dynamic tunnel. A value 0 indicates that the dynamic tunnel is protecting all source ports.

AssociatedFiltDestPort
Indicates the destination port number of the data traffic that is being protected by the dynamic tunnel. A value 0 indicates that the dynamic tunnel is protecting all destination ports.

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 value of State is `PENDING`. Otherwise, this field has a 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; a value of 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: IP address in the source NAT-OA payload 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-OA payload was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a source NAT-OA payload. For information about accessing RFCs, see Appendix F, "Related protocol specifications," on page 891.

For a non-UDP-encapsulated transport SA: The value is n/a.

DstNATOARcvd
For a UDP-encapsulated transport SA: IP address in the destination NAT-OA payload 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 host's public address. This value is 0.0.0.0 if a destination NAT-OA payload was not received. An IKE peer at a pre-RFC3947 NAT Traversal support level cannot send a destination NAT-OA payload. For information about accessing RFCs, see Appendix F, "Related protocol specifications," on page 891.

For a non-UDP-encapsulated transport SA: The value is n/a.

LclIpSpecExIDPayload
The local IP specification exchanged in the ID payloads or 0 if no ID payloads were exchanged. 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 in the ID payloads or 0 if no ID payloads were exchanged. The remote IP specification can be a single IP address, IP address range, IP address followed by 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 891.

Interface (-i) primary option

Purpose

The -i primary option is used to display interface information that is defined to the specified TCPIP stack. Interface configuration is obtained from the TCPIP profile. This interface display applies only to stacks that are configured with IPSECURITY.

See “Interface (-i) option” on page 552 for parameter descriptions.

Syntax

For -i primary option syntax see “The z/OS UNIX ipsec command syntax” on page 541.

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.

Report examples

`ipsec -i display`

```
CS V1R9 ipsec  Stack Name: TCPCS1    Mon Jun 14 14:48:50 2004
Primary:  Interface       Function: display           Format:   detail
Source:   Stack           Scope:    n/a               TotAvail: 57

InterfaceName                 VIPA1L
SecurityClass                 000
Active                        YES
DVIPA                         NO
Address                       10.5.4.1
*************************************************************************
InterfaceName                 NSQDIO1L
SecurityClass                 255
Active                        NO
DVIPA                         NO
Address                       10.1.1.1
*************************************************************************
InterfaceName                 VIPL0A0B0B01
SecurityClass                 000
Active                        YES
DVIPA                         YES
Address                       10.11.11.1
*************************************************************************
InterfaceName                 NSQDIO16
SecurityClass                 006
Active                        YES
DVIPA                         NO
Address                       2001::10:5:3:4
```

Report field descriptions

For more information about the header, see “Report heading” on page 558.

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

Purpose

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 “IP traffic test (-t) option” on page 552 for parameter descriptions.

Syntax

For -t primary option syntax see “The z/OS UNIX ipsec command syntax” on page 541.

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.

Report example

ipsec -t 2001:db8:10::81:2:6 2001:db8:10::81:8:6 tcp 1027 21 out

```
CS V1R9 ipsec  Stack Name: TCPCS4  Fri Feb  3 12:32:55 2006
Primary:  IP Traffic Test Function: Display            Format:   Detail
Source:   Stack Policy    Scope:    n/a                TotAvail: 4
TestData: 2001:db8:10::81:2:6  2001:db8:10::81:8:6  tcp 1027 21 out

FilterName:                  GW-GW-sub2range-v6~7
FilterNameExtension:         1
GroupName:                   n/a
LocalStartActionName:        GW-GW-sub2range-v6~6
VpnActionName:               Hmac_Md5-DES
TunnelID:                    Y24
Type:                        Dynamic
State:                       Active
Action:                      Permit
Scope:                       Routed
Direction:                   Outbound
OnDemand:                    Yes
SecurityClass:               0
Logging:                     All
Protocol:                    TCP(6)
ICMPType:                    n/a
ICMPCode:                    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
***********************************************************************
FilterName:                  GW-GW-sub2range-v6~7
FilterNameExtension:         1
GroupName:                   n/a
LocalStartActionName:        GW-GW-sub2range-v6~6
VpnActionName:               Hmac_Md5-DES
TunnelID:                    Y0
Type:                        Dynamic Anchor
State:                       Active
Action:                      Permit
Scope:                       Routed
Direction:                   Outbound
OnDemand:                    Yes
SecurityClass:               0
Logging:                     All
Protocol:                    TCP(6)
ICMPType:                    n/a
ICMPCode:                    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
***********************************************************************
FilterName:                   Range2all-v6~6
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         Range2all-v6~5
VpnActionName:                Hmac_SHA-AES
TunnelID:                     Y0
Type:                         Dynamic Anchor
State:                        Active
Action:                       Permit
Scope:                        Routed
Direction:                    Outbound
OnDemand:                     Yes
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     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:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  ::
DestAddressPrefix:            0
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
***********************************************************************
FilterName:                   DenyAllRule_Generated__________Outbnd_v6
FilterNameExtension:          n/a
GroupName:                    n/a
LocalStartActionName:         n/a
VpnActionName:                n/a
TunnelID:                     0x00
Type:                         Generic
State:                        Active
Action:                       Deny
Scope:                        Both
Direction:                    Outbound
OnDemand:                     n/a
```

```
SecurityClass:              0
Logging:                    None
Protocol:                   All
ICMPType:                   n/a
ICMPCode:                   n/a
OSPFType:                   n/a
TCPQualifier:               n/a
ProtocolGranularity:        Rule
SourceAddress:              ::
SourceAddressPrefix:        0
SourceAddressRange:         n/a
SourceAddressGranularity:   Packet
SourcePort:                 All
SourcePortRange:            n/a
SourcePortGranularity:      Rule
DestAddress:                ::
DestAddressPrefix:          0
DestAddressRange:           n/a
DestAddressGranularity:     Packet
DestPort:                   All
DestPortRange:              n/a
DestPortGranularity:        Rule
OrigRmtConnPort:            n/a
RmtIDPayload:               n/a
RmtUdpEncapPort:            n/a
**************************************************
```

ipsec -t 9.42.105.144 9.42.105.3 tcp 0 0

```
CS V1R9 ipsec  Stack Name: TCPCS  Wed Sep 15 20:37:06 2004
Primary:  IP Traffic Test Function: Display           Format:   Detail
Source:   Stack Policy    Scope:    n/a               TotAvail: 3
TestData: 9.42.105.144  9.42.105.3  tcp 0 0

FilterName:                 LocalNtt_Log
FilterNameExtension:        1
GroupName:                  n/a
LocalStartActionName:       LocalStartActNtt
VpnActionName:              DynVpnAct
TunnelID:                   Y2
Type:                       NATT Dynamic
State:                      Active
Action:                     Permit
Scope:                      Local
Direction:                  Outbound
OnDemand:                   No
SecurityClass:              0
Logging:                    All
Protocol:                   TCP(6)
ICMPType:                   n/a
ICMPCode:                   n/a
OSPFType:                   n/a
TCPQualifier:               None
ProtocolGranularity:        Rule
SourceAddress:              9.42.105.144
SourceAddressPrefix:        n/a
SourceAddressRange:         n/a
SourceAddressGranularity:   Packet
SourcePort:                 All
SourcePortRange:            n/a
SourcePortGranularity:      Rule
DestAddress:                9.42.105.3
DestAddressPrefix:          n/a
DestAddressRange:           n/a
DestAddressGranularity:     Packet
DestPort:                   All
DestPortRange:              n/a
DestPortGranularity:        Rule
```

```
OrigRmtConnPort:              n/a
RmtIDPayload:                 192.168.90.1
RmtUdpEncapPort:              4500
************************************************************************
FilterName:                   LocalNtt_Log
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         LocalStartActNtt
VpnActionName:                DynVpnAct
TunnelID:                     Y0
Type:                         NATT Anchor
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     No
SecurityClass:                0
Logging:                      All
Protocol:                     TCP(6)
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 None
ProtocolGranularity:          Rule
SourceAddress:                9.42.105.144
SourceAddressPrefix:          n/a
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
SourcePortRange:              n/a
SourcePortGranularity:        Rule
DestAddress:                  9.42.105.3
DestAddressPrefix:            n/a
DestAddressRange:             n/a
DestAddressGranularity:       Packet
DestPort:                     All
DestPortRange:                n/a
DestPortGranularity:          Rule
OrigRmtConnPort:              n/a
RmtIDPayload:                 n/a
RmtUdpEncapPort:              n/a
************************************************************************
FilterName:                   LocalNtt_Log
FilterNameExtension:          1
GroupName:                    n/a
LocalStartActionName:         LocalStartActNtt
VpnActionName:                DynVpnAct
TunnelID:                     Y0
Type:                         Dynamic Anchor
State:                        Active
Action:                       Permit
Scope:                        Local
Direction:                    Outbound
OnDemand:                     No
SecurityClass:                0
Logging:                      All
Protocol:                     All
ICMPType:                     n/a
ICMPCode:                     n/a
OSPFType:                     n/a
TCPQualifier:                 n/a
ProtocolGranularity:          Rule
SourceAddress:                9.42.105.0
SourceAddressPrefix:          24
SourceAddressRange:           n/a
SourceAddressGranularity:     Packet
SourcePort:                   All
```

```
SourcePortRange:             n/a
SourcePortGranularity:       Rule
DestAddress:                 9.42.105.3
DestAddressPrefix:           n/a
DestAddressRange:            n/a
DestAddressGranularity:      Packet
DestPort:                    All
DestPortRange:               n/a
DestPortGranularity:         Rule
OrigRmtConnPort:             n/a
RmtIDPayload:                n/a
RmtUdpEncapPort:             n/a
***********************************************************************

3 entries selected
```

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 “Report heading” on page 558.

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 560 for a description of the fields in each filter.

NATT port translation (-o) primary option

Purpose

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 “NATT port translation (-o) option” on page 553 for parameter descriptions.

Syntax

For -o primary option syntax see “The z/OS UNIX ipsec command syntax” on page 541.

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.

Report examples

ipsec -o display

```
CS V1R9 ipsec  Stack Name: TCPCS  Thu Aug 26 05:00:02 2004
Primary:  NATT Port Trans Function: Display           Format:   Detail
```

```
Source:   Stack           Scope:    Current            TotAvail: 4

RmtIpAddress:         9.42.105.3
Protocol:            TCP(6)
TransRmtConnPort:     3005
OrigRmtConnPort:      3005
RmtInnerIpAddress:    192.168.90.1
***********************************************************************
RmtIpAddress:         9.42.105.3
Protocol:            TCP(6)
TransRmtConnPort:     3006
OrigRmtConnPort:      3006
RmtInnerIpAddress:    192.168.90.1
***********************************************************************
RmtIpAddress:         9.42.105.3
Protocol:            UDP(17)
TransRmtConnPort:     3003
OrigRmtConnPort:      3003
RmtInnerIpAddress:    192.168.90.1
***********************************************************************
RmtIpAddress:         9.42.105.3
Protocol:            UDP(17)
TransRmtConnPort:     65535
OrigRmtConnPort:      3003
RmtInnerIpAddress:    192.168.90.2
***********************************************************************

4 entries selected
```

Report field descriptions

For the header information, see “Report heading” on page 558.

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 peer’s private address from the source NAT-OA payload received during the IKE negotiation.
- Otherwise, this field displays the value `n/a`.

IKED network security information (-w) primary option

Purpose

The -w primary option is used to display network security configuration information for each active stack on the system.

See “IKED network security (-w) option” on page 554 for parameter descriptions.

Syntax

For -w primary option syntax see “The z/OS UNIX ipsec command syntax” on page 541.

Command syntax examples

ipsec -w display
: Displays network security information of each active stack on the system.

Report examples

```
ipsec -w display

CS V1R9 ipsec   Stack Name: n/a  Mon Jun 14 14:48:50 2004
Primary:  Stack NS        Function: display          Format:     detail
Source:   IKED            Scope: n/a                 TotAvail:   n/a
SystemName:  zsystem4

StackName:                    tcpcs1
ClientName:                   nsclient1
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
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
```

Report field descriptions

For the header information, see “Report heading” on page 558.

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*.

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

Purpose

The -x primary option is used to display information about NSS clients that are currently connected to the NSS server.

See “Network security server (-x) option” on page 554 for parameter descriptions.

Syntax

For -x primary option syntax see “The z/OS UNIX ipsec command syntax” on page 541.

Command syntax examples

ipsec -x display
Display the status of all 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.

Report examples

`ipsec -x display`

```
CS V1R9 ipsec   NS Client Name: n/a  Mon Jun 14 14:48:50 2004
Primary:  NS Server       Function: display         Format:     detail
Source:   Server          Scope: n/a                TotAvail:   n/a
System Name: zsystem7

ClientName:                  client1
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
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
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
```

Report field descriptions

For the header information, see “Report heading” on page 558.

ClientName
The name of the NSS client.

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.

Chapter 5. Displaying policy-based networking information

This information describes how to 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 247 for Netstat commands, such as NETSTAT SLAP (onetstat -j) and additional information that might be relevant to retrieving information from the network and “ipsec command security” on page 538 for 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 Network SLAPM2 subagent information in the *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

Purpose

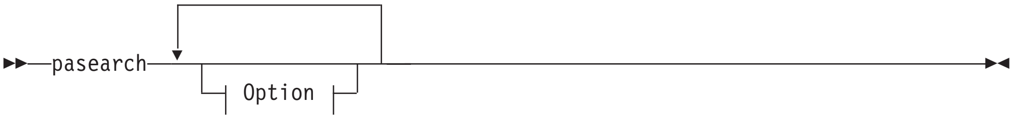
Use the z/OS UNIX pasearch command to query information from the z/OS UNIX Policy Agent. The command is issued from the UNIX System Services shell.

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 the *z/OS Communications Server: IP Configuration Guide* for information on 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 again 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.

The following are 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 will be 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 should be 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 for the last update for this policy type.

-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 V1R9                   Image Name: TCPCS
  Date:                 07/31/2006        Time:  15:33:40
  PAPI Version          6                 DLL Version:  6

Qos Policy Object:
  ConfigLocation:       Local             LDAPServer:        False
  ImageFileName:        /u/user10/pagallqos.conf
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     True
  AtomicParse:          False             DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   False             ModifyOnIDChange:  True
  Configured:           True              UpdateInterval:    1800
  PerfColEnabled:       False
  InstanceId:           1154374260
  LastPolicyChanged:    Mon Jul 31 15:31:00 2006
```

```
Ids Policy Object:
  ConfigLocation:       Local             LDAPServer:        False
  CommonFileName:       /u/user10/pagallcommonids.conf
  ImageFileName:        /u/user10/pagallids.conf
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     True
  AtomicParse:          False             DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   False             ModifyOnIDChange:  False
  Configured:           True              UpdateInterval:    1800
  InstanceId:           1154374260
  LastPolicyChanged:    Mon Jul 31 15:31:00 2006


IPSec Policy Object:
  ConfigLocation:       Remote            LDAPServer:        False
  ClientName:           client_nineteen
  ClientUserid:         USER10
  PolicyServerAddr      9.42.104.23
  PolicyServerPort:     8211              PolicyServSysname: VIC137
  ConnectTime:          Mon Jul 31 15:31:01 2006
  ApplyFlush:           False
  ApplyPurge:           False
  AtomicParse:          True              DeleteOnNoflush:   True
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  IpSecEnabled IPv4:    True              IpSecEnabled IPv6: False
  IpSec3DESEnabled:     True              IpSecAESEnabled:   True
  UpdateInterval:       3600
  InstanceId:           1154374265
  LastPolicyChanged:    Mon Jul 31 15:31:05 2006

  IpFilter Policy Object:
   Configured:          True              PreDecapOn:        Off
   FilterLogging:       On                FilterLogImplicit: No
   AllowOnDemand:       No
  KeyExchange Policy Object:
   Configured:          True
   AllowNat:            No                NatKeepAliveIntvl: 20
  LocalDynVpn Policy Object:
   Configured:          True

Routing Policy Object:
  ConfigLocation:       Local             LDAPServer:        FALSE
  CommonFileName:       pagrout.common
  ImageFileName:        pagrout.routing
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     False
  AtomicParse:          True              DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  Configured:           False             UpdateInterval:    3600

TTLS Policy Object:
  ConfigLocation:       Remote            LDAPServer:        False
  ClientName:           client_nineteen
  ClientUserid:         USER10
  PolicyServerAddr      9.42.104.23
  PolicyServerPort:     8211              PolicyServSysname: VIC137
  ConnectTime:          Mon Jul 31 15:31:01 2006
  ApplyFlush:           True              PolicyFlush:       True
  ApplyPurge:           True              PurgePolicies:     False
  AtomicParse:          True              DeleteOnNoflush:   False
  DummyOnEmptyPolicy:   True              ModifyOnIDChange:  False
  Configured:           True              UpdateInterval:    3600
  InstanceId:           1154374265
  LastPolicyChanged:    Mon Jul 31 15:31:05 2006
```

The following example shows active QoS policies for TCP image TCPCS:

```
pasearch -q -p TCPCS

TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 10/21/2003        Time:  08:38:43
  QoS Instance Id:      1066739595

policyRule:             disttelnet-rule
  Rule Type:            QoS
  Version:              3                 Status:            Active
  Distinguish Name:     cn=disttelnet-rule,cn=QoS,cn=advanced,ou=policy,o=IBM,c=US
  Group Distinguish Nm: cn=sysplex,cn=QoS,cn=advanced,ou=policy,o=IBM,c=US
  Weight:               120               ForLoadDist:       True
  Priority:             20                Sequence Actions:  Don't Care
  No. Policy Action:    1
  policyAction:         telnetGold-action
   ActionType:          QOS
   Action Sequence:     1
  Time Periods:
   Day of Month Mask:   1111111111111111111111111111111
   Month of Yr Mask:    111111111111
   Day of Week Mask:    0111110  (Sunday - Saturday)
   Start Date Time:     Sat Jul  1 00:00:00 2000

   Start Date Time UTC:  Sat Jul  1 04:00:00 2000

   End Date Time:       Fri Jul  1 00:00:00 2005

   End Date Time UTC:    Fri Jul  1 04:00:00 2005

   Fr TimeOfDay:        06:00             To TimeOfDay:      22:00
   Fr TimeOfDay UTC:    10:00             To TimeOfDay UTC:  02:00
   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:
   UserCondition:
    UserName:
    GroupName:
   ApplicationCondition:
    ProtocolNumFrom:    6                 ProtocolNumTo:     6
    SourcePortFrom:     0                 SourcePortTo:      0
    DestPortFrom:       23                DestPortTo:        23
    ApplicationName:                      ApplPriority:      0
    ApplicationData:

  Qos Action:           telnetGold-action
    Version:            3                 Status:            Active
    Distinguish Name:   cn=telnetGold,cn=QoSact,cn=repository,o=IBM,c=US
    Scope:              DataTraffic       OutgoingTOS:       10100000
    Permission:         Allowed
    MaxRate:            0                 MinRate:           500
    MaxConn:            0
    Routing Interfaces: 4
      Interface  1:          129.100.11.1
      Interface  2:          129.100.21.1
      Interface  3:          129.200.12.1
      Interface  4:          0.0.0.0
    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
    Inbound Connection Attributes:
     InboundScope:      Connection        AvgConnRate:       100
     ConnBurstSize:     5                 PeakConnRate:      0
     AvgApplReqRate:    100               ApplReqBurstSize:  5
     ApplReqPeakRate:   0                 PrioritizedQueue:  2
```

The following example shows active KeyExchange policies:

pasearch -v k

```
TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 01/11/2005        Time:  10:52:32
  IPSec Instance Id:    1105458730

policyRule:             Keyex_1
  Rule Type:            KeyExchange
  Version:              3                 Status:            Active
  GroupName:            Keyex_Group_1
  Weight:               102               ForLoadDist:       False
  Priority:             2                 Sequence Actions:  Don't Care
  No. Policy Action:    1
  IpSecType:            policyKeyExchange
  policyAction:         Keyact_1
   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:          77.77.77.77
     Identity:
      x500dn:
       CN=John Smith,OU=" Raw Materials ",OU=Purchasing Dept,O=Acme Manufacturing,ST=NY,C=usa
    RemoteSecurityEndPoint:
     Location:
      FromAddr:         66.66.66.66
      Prefix:           1
     Identity:
      Fqdn:
       jkfranks.raleigh.ibm.com
     CaLabel:           Label99
     CaLabel:           Label98
     CaLabel:           Label97

  KeyExchange Action:   Keyact_1
    Version:            3                 Status:            Active
    HowToInitiate:      DoNot             HowToRespond:      Main
    AllowNat            Yes
    KeyExchangeOffer:   0
     HowToEncrypt:      DES               HowToAuthMsgs:     SHA1
     HowToAuthPeers:    RsaSignature      DHGroup:           Group2
     RefLifeTmPropose:  21
     RefLifeTmAcptMin:  20                RefLifeTmAcptMax:  30
     RefLifeSzPropose:  51
     RefLifeSzAcptMin:  50                RefLifeSzAcptMax:  60
    KeyExchangeOffer:   1
     HowToEncrypt:      DES               HowToAuthMsgs:     SHA1
     HowToAuthPeers:    RsaSignature      DHGroup:           Group1
     RefLifeTmPropose:  2001
     RefLifeTmAcptMin:  2000              RefLifeTmAcptMax:  3000
     RefLifeSzPropose:  5001
     RefLifeSzAcptMin:  5000              RefLifeSzAcptMax:  6000
    KeyExchangeOffer:   2
     HowToEncrypt:      DES               HowToAuthMsgs:     MD5
     HowToAuthPeers:    PresharedKey      DHGroup:           Group1
     RefLifeTmPropose:  201
     RefLifeTmAcptMin:  200               RefLifeTmAcptMax:  300
     RefLifeSzPropose:  501
     RefLifeSzAcptMin:  500               RefLifeSzAcptMax:  600
```

The following example shows an active LocalDynVpn policy rule:

pasearch -v l

```
TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 01/11/2005        Time:  10:52:32
  IPSec Instance Id:    1105458730

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:          9.1.1.1
     ToAddr:            9.1.1.1
    RemoteIp:
     FromAddr:          10.3.0.0
     Prefix:            16
    LocalDataPort:      12000             RemoteDataPort:    12000
    AutoActivate:       Yes
    Protocol:           UDP  (17)
```

The following example shows all active IPSec policies names only:

pasearch -v a -n

```
TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 01/11/2005        Time:  10:52:32
  IPSec Instance Id:    1105458730

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:             Rule1All-Permit
  IpFilter Action:      permit

policyRule:             Rule2All-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:             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 V1R9                   Image Name: TCPCS2
  Date:                 11/03/2006        Time:  08:15:55
  IPSec Instance Id:    1162556092

policyRule:             Rule2Admin
  Rule Type:            IpFilter
  Version:              3                 Status:            Active
  GroupName:            Admin
  Weight:               113               ForLoadDist:       False
  Priority:             13                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
  Condition Work Level:      0
    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
  IpSec Condition Work:                   NegativeIndicator: Off
   IpFilter Condition:
    Source Address:
     FromAddr:          10.1.1.1
     ToAddr:            10.1.1.1
    Destination Address:
```

```
    Service Condition:
     Protocol:         0
     Direction:        0
     RouteType:        0                 SecurityClass:     0
  Condition Work Level:      1
    Group Number:      2                 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
  IpSec Condition Work:                  NegativeIndicator: Off
   IpFilter Condition:
    Source Address:
    Destination Address:
     FromAddr:         10.1.1.2
     ToAddr:           10.1.1.2
    Service Condition:
     Protocol:         0
     Direction:        0
     RouteType:        0                 SecurityClass:     0
  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
  IpSec Condition Work:                  NegativeIndicator: Off
   IpFilter Condition:
    Source Address:
    Destination Address:
    Service Condition:
     Protocol:         All
     Direction:        Bidirectional
     RouteType:        Local             SecurityClass:     0

  IpFilter Action:     ipsec
    Version:           3                 Status:            Active
    Scope:             GenericFilter
    ipFilterAction:    IPSec             IpFilterLogging:   Yes Logdeny

  IpFilter Action:     Silver-TransportMode
    Version:           3                 Status:            Active
    Scope:             DynamicVpn
    Initiation:        Either            VpnLife:           1440
    AcceptablePfs:     None
    InitiateWithPfs:   None              IpDataOfferNum:    1
    IPDataOffer:       0
     HowToEncap:       Transport         HowToEncrypt:      DES
     HowToAuth:        ESP               HowToAuthAlgr:     HMAC_SHA
     RefLifeTmPropose: 240
     RefLifeTmAcptMin: 120               RefLifeTmAcptMax:  480
     RefLifeSzPropose: None
     RefLifeSzAccept : None
```

The following example shows active IDS policies whose names match the prefix AttackMalformed:

```
pasearch -i -w -f AttackMalformed
```

```
TCP/IP pasearch CS V1R9                  Image Name: TCPCS
  Date:                08/02/2005        Time:  12:56:02
  IDS Instance Id:     1123001734

policyRule:            AttackMalformed-rule
  Rule Type:           IDS
  Version:             3                 Status:            Active
  Distinguish Name:    cn=attackMalformed-rule,cn=IDS,cn=starter,ou=policy,o=IBM,c=US
  Group Distinguish Nm: cn=IDS,cn=starter,ou=policy,o=IBM,c=US
  Weight:              102               ForLoadDist:       False
  Priority:             2                 Sequence Actions:  Don't Care
  No. Policy Action:    1
  IdsType:              policyIdsAttack
  policyAction:         AttackLog-action
   ActionType:          IDS
   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
  Ids Condition Summary:                  NegativeIndicator: Off
   AttackCondition:
    IdsAttackType:      malformed_packet
    IPOptionFrom:       0                 IPOptionTo:        0
   TransportCondition:
    LocalPortFrom:      0                 LocalPortTo:       0
    RemotePortFrom:     0                 RemotePortTo:      0
    ProtocolNumFrom:    0                 ProtocolNumTo:     255
   HostCondition:
    LocalIpAddrFrom:    0.0.0.0           LocalIpAddrTo:     0.0.0.0
    RemoteIpAddrFrom:   0.0.0.0           RemoteIpAddrTo:    0.0.0.0

  Ids Action:           AttackLog-action
    Version:            3                 Status:            Active
    IdsType:            policyIdsAttack
    Distinguish Name:   cn=attackact1,cn=IDSact,cn=repository,o=IBM,c=US
    Notification Attributes:
     Notification:      Syslog
     TraceData:         RecordSize
     TypeActions:       Log ExceptStats
     StatInterval:      60                LoggingLevel:      1
     TraceRecordSize:   200
    Attack Actions Attributes:
     MaxEventMessage:   0
     IfcFloodPercent:   10                IfcFloodMinDisc:   1000
```

The following example shows active IDS rules and actions configured from the IDS configuration file:

pasearch -i

```
TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 11/30/2005        Time:  12:14:14
  IDS Instance Id:      1133370849

policyRule:             ScanEventLowTcp-rule
  Rule Type:            IDS
  Version:              4                 Status:            Active
  Weight:               2                 ForLoadDist:       False
  Priority:             0                 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:    05:00             To TimeOfDay UTC:  05:00
   TimeZone:            Local
  Ids Condition Summary:                  NegativeIndicator: Off
   ScanEvent Condition:
    Sensitivity:        Low
    Protocol:           TCP  (6)
    LocalPortFrom:      1                 LocalPortTo:       1023
    LocalHostAddress:
     FromAddr:          All
     ToAddr:            All

  Ids Action:           ScanEventLow-action
    Version:            4                 Status:            Active
    ActionType:         ScanEvent         ScanEventType:     Count
    TypeActions:        0
    statType:           Normal            StatInterval:      60
    TraceData:          Header            TraceRecordSize:   100
    LogDetail:          No                LoggingLevel:      4
    MaxEventMessage:    0
```

The following example shows active AT-TLS policies:

pasearch -t

```
TCP/IP pasearch CS V1R9                   Image Name: TCPCS
  Date:                 09/27/2004        Time:  09:17:24
  TTLS Instance Id:     1096291029

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_act1
   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

  TTLS Action:                  grp_act1
    Version:                    3
    Status:                     Active
    Scope:                      Group
    TTLSEnabled:                On
    CtraceClearText:            Off
    Trace:                      6
    TTLSGroupAdvancedParms:
     SecondaryMap:              Off
     SyslogFacility:            Daemon
     EnvFile:                   /u/user140/EnvParms
    GroupUserInstance:          3

  TTLS Action:                  Secure_Telnet_Env
    Version:                    3
    Status:                     Active
    Scope:                      Environment
    HandshakeRole:              Server
    TTLSKeyringParms:
     Keyring:                   TCPCSsafkeyring
    TTLSEnvironmentAdvancedParms:
     SSLv2:                     Off
     SSLv3:                     On
     TLSv1:                     On
     ApplicationControlled:     Off
     HandshakeTimeout:          5
     ClientAuthType:            Required
     ResetCipherTimer:          0
    EnvironmentUserInstance:    0

  TTLS Action:                  Secure_Telnet_Conn_Debug
    Version:                    3
    Status:                     Active
    Scope:                      Connection
    CtraceClearText:            On
    Trace:                      254
```

The following example shows active routing policies:

```
pasearch -R
TCP/IP pasearch CS V1R9                   Image Name: TCPCS2
  Date:                 11/02/2006        Time:  11:36:03
  Routing Instance Id:  1162481223

policyRule:             EE-RoutingRule
  Rule Type:            Routing
  Version:              4                 Status:            Active
  Weight:               100               ForLoadDist:       False
  Priority:             100               Sequence Actions:  Don't Care
  No. Policy Action:    1
  policyAction:         EE-RoutingAction
   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:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Routing Condition Summary:              NegativeIndicator: Off
   IpSourceAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   IpDestAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   TrafficDescriptor:
    Protocol:           UDP  (17)
    SourcePortFrom      12000             SourcePortTo       12004
    DestinationPortFrom 12000             DestinationPortTo  12004
    JobName                               SecurityZone
    SecurityLabel

  Routing Action:       EE-RoutingAction
    Version:            4                 Status:            Active
    UseMainRouteTable   No
    RouteTable:         EERteTbl

policyRule:             FTP-RoutingRule
  Rule Type:            Routing
  Version:              4                 Status:            Active
  Weight:               90                ForLoadDist:       False
  Priority:             90                Sequence Actions:  Don't Care
  No. Policy Action:    1
  policyAction:         FTP-RoutingAction
   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:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Routing Condition Summary:              NegativeIndicator: Off
   IpSourceAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   IpDestAddr Address:
    FromAddr:           1.1.1.1
    ToAddr:             1.1.1.1
   TrafficDescriptor:
    Protocol:           TCP  (6)
    SourcePortFrom      0                 SourcePortTo       0
    DestinationPortFrom 0                 DestinationPortTo  0
    JobName             FTP*              SecurityZone
    SecurityLabel

  Routing Action:       FTP-RoutingAction
    Version:            4                 Status:            Active
    UseMainRouteTable   Yes
    RouteTable:         FTPRtTbl

policyRule:             SecZone1-RoutingRule
  Rule Type:            Routing
  Version:              4                 Status:            Active
  Weight:               80                ForLoadDist:       False
  Priority:             80                Sequence Actions:  Don't Care
  No. Policy Action:    1
  policyAction:         SecZone1-RoutingAction
   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:        00:00             To TimeOfDay:      24:00
   Fr TimeOfDay UTC:    04:00             To TimeOfDay UTC:  04:00
   TimeZone:            Local
  Routing Condition Summary:              NegativeIndicator: Off
   IpSourceAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   IpDestAddr Address:
    FromAddr:           0.0.0.0
    Prefix:             0
   TrafficDescriptor:
    Protocol:           All
    SourcePortFrom      5000              SourcePortTo       5000
    DestinationPortFrom 0                 DestinationPortTo  0
    JobName                               SecurityZone       SECZONE1
    SecurityLabel

  Routing Action:       SecZone1-RoutingAction
```

```
    Version:            4                 Status:            Active
    UseMainRouteTable   No
    RouteTable:         Zone1RT

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:    12:00             To TimeOfDay UTC:  21:00
   TimeZone:            Local
  Routing Condition Summary:              NegativeIndicator: Off
   IpSourceAddr Address:
    FromAddr:           1.0.0.1
    ToAddr:             1.0.0.1
   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

  Routing Action:       GenericRoutingAction
    Version:            4                 Status:            Active
    UseMainRouteTable   Yes
    RouteTable:         RtTbl1
    RouteTable:         RtTbl2
    RouteTable:         RtTbl3
```

The following example shows active route tables:

```
pasearch -T
TCP/IP pasearch CS V1R9                   Image Name: TCPCS2
  Date:                 11/02/2006        Time:  11:36:17
  Routing Instance Id:  1162481223

  Route Table:          EERteTbl
    Version:            1                 Status:            Active
    IgnorePathMtuUpdate No                Multipath          Disable
    DynamicXCFRoutes    No
    DynamicRoutingParms
     link_name             OSALINK2
     gateway_addr          10.11.12.1
    DynamicRoutingParms
     link_name             OSALINK3
     gateway_addr          10.11.13.1

  Route Table:          FTPRtTbl
    Version:            1                 Status:            Active
    IgnorePathMtuUpdate No                Multipath          UseGlobal
    DynamicXCFRoutes    No
    Route
     Destination           DEFAULT
     First Hop:
      gateway_addr         10.1.1.1
      link_name            LARGEMTULINK
     MTU size              4096
     Replaceable           No
     MaximumRetransmitTime 120.000
     MinimumRetransmitTime 0.500
     RoundTripGain         0.125
     VarianceGain          0.250
     VarianceMultiplier    2.000
     DelayAcks             Yes

  Route Table:          RtTbl1
    Version:            1                 Status:            Active
    IgnorePathMtuUpdate No                Multipath          PerConnection
    DynamicXCFRoutes    No
    Route
     Destination:
      ipaddress            1.1.1.1
     First Hop:
      gateway_addr         =
      link_name            LINK1
     MTU size              4096
     Replaceable           No
     MaximumRetransmitTime 120.000
     MinimumRetransmitTime 0.500
     RoundTripGain         0.125
     VarianceGain          0.250
     VarianceMultiplier    2.000
     DelayAcks             Yes
    Route
     Destination:
      ipaddress            1.1.0.0
      Prefix               16
     First Hop:
      gateway_addr         2.2.2.2
      link_name            LINK2
     MTU size              4096
     Replaceable           No
     MaximumRetransmitTime 120.000
     MinimumRetransmitTime 0.500
     RoundTripGain         0.125
     VarianceGain          0.250
     VarianceMultiplier    2.000
     DelayAcks             Yes
    Route
     Destination:
      ipaddress            1.1.1.0
      Prefix               24
     First Hop:
      gateway_addr         2.2.2.2
      link_name            LINK2
     MTU size              4096
     Replaceable           No
     MaximumRetransmitTime 120.000
     MinimumRetransmitTime 0.500
     RoundTripGain         0.125
     VarianceGain          0.250
     VarianceMultiplier    2.000
     DelayAcks             Yes
    Route
     Destination:
      ipaddress            254.0.0.0
      Prefix               7
     First Hop:
```

```
  gateway_addr         3.3.3.3
  link_name            LINK3
 MTU size              DEFAULTSIZE
 Replaceable           No
 MaximumRetransmitTime 120.000
 MinimumRetransmitTime 0.500
 RoundTripGain         0.125
 VarianceGain          0.250
 VarianceMultiplier    2.000
 DelayAcks             Yes
Route
 Destination           DEFAULT
 First Hop:
  gateway_addr         4.4.4.4
  link_name            LINK4
 MTU size              4096
 Replaceable           No
 MaximumRetransmitTime 120.000
 MinimumRetransmitTime 0.500
 RoundTripGain         0.125
 VarianceGain          0.250
 VarianceMultiplier    2.000
 DelayAcks             Yes
Route
 Destination:
  ipaddress            1.2.2.2
 First Hop:
  gateway_addr         2.2.2.2
  link_name            LINK2
 MTU size              4096
 Replaceable           No
 MaximumRetransmitTime 120.000
 MinimumRetransmitTime 0.500
 RoundTripGain         0.125
 VarianceGain          0.250
 VarianceMultiplier    2.000
 DelayAcks             Yes
Route
 Destination:
  ipaddress            1.2.2.2
 First Hop:
  gateway_addr         3.3.3.3
  link_name            LINK3
 MTU size              DEFAULTSIZE
 Replaceable           No
 MaximumRetransmitTime 120.000
 MinimumRetransmitTime 0.500
 RoundTripGain         0.125
 VarianceGain          0.250
 VarianceMultiplier    2.000
 DelayAcks             Yes
Route
 Destination:
  ipaddress            1.2.2.2
 First Hop:
  gateway_addr         4.4.4.4
  link_name            LINK4
 MTU size              4096
 Replaceable           No
 MaximumRetransmitTime 120.000
 MinimumRetransmitTime 0.500
 RoundTripGain         0.125
 VarianceGain          0.250
 VarianceMultiplier    2.000
 DelayAcks             Yes
```

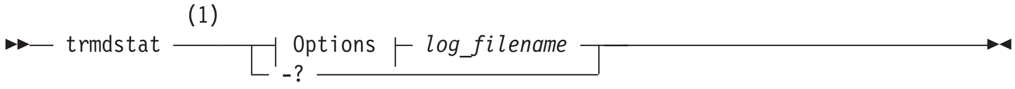
```
Route Table:          RtTbl2
  Version:            1                 Status:            Active
  IgnorePathMtuUpdate No                Multipath          UseGlobal
  DynamicXCFRoutes    No
  DynamicRoutingParms
   link_name             LINK1
  DynamicRoutingParms
   link_name             LINK2
   gateway_addr          2.1.1.1
  DynamicRoutingParms
   link_name             LINK2
   gateway_addr          2.2.2.2

Route Table:          RtTbl3
  Version:            1                 Status:            Active
  IgnorePathMtuUpdate No                Multipath          UseGlobal
  DynamicXCFRoutes    No
  Route
   Destination:
    ipaddress            1.1.1.1
   First Hop:
    gateway_addr         =
    link_name            LINK1
   MTU size              4096
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  Route
   Destination:
    ipaddress            1.1.0.0
    Prefix               16
   First Hop:
    gateway_addr         2.2.2.2
    link_name            LINK2
   MTU size              4096
   Replaceable           Yes
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
  DynamicRoutingParms
   link_name             LINK2

Route Table:          Zone1RT
  Version:            1                 Status:            Active
  IgnorePathMtuUpdate No                Multipath          UseGlobal
  DynamicXCFRoutes    No
  Route
   Destination           DEFAULT
   First Hop:
    gateway_addr         10.2.2.2
    link_name            SECLINK
   MTU size              4096
   Replaceable           No
   MaximumRetransmitTime 120.000
   MinimumRetransmitTime 0.500
   RoundTripGain         0.125
   VarianceGain          0.250
   VarianceMultiplier    2.000
   DelayAcks             Yes
```

The z/OS UNIX trmdstat command—Display traffic regulation management daemon (TRMD) log

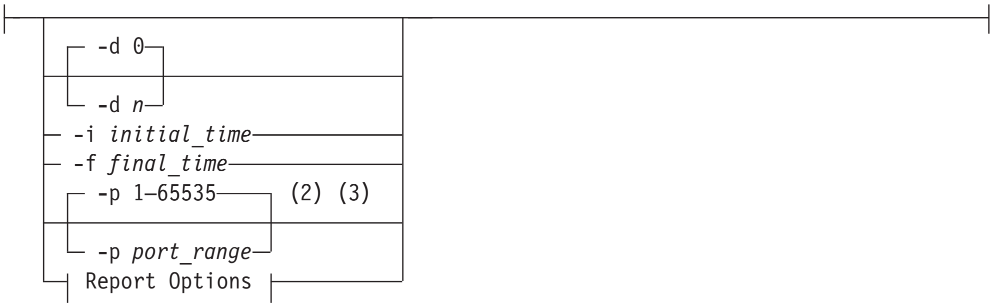
Purpose

Use the trmdstat to give a consolidated view of the log messages written out by the Traffic Regulation Management daemon (TRMD).

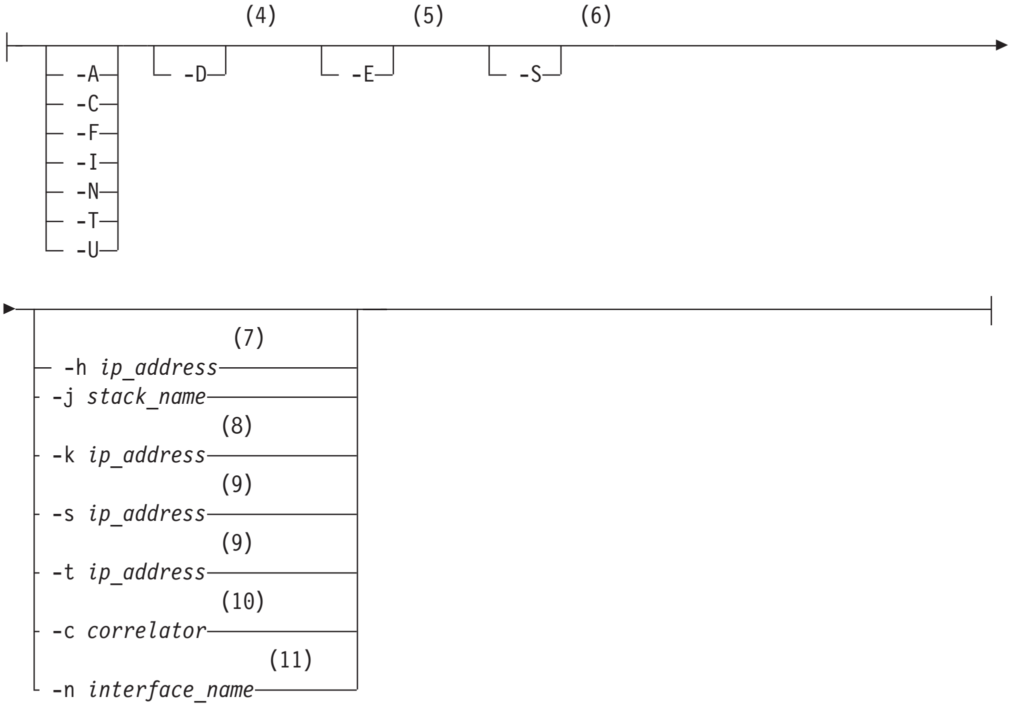
Format



Options:



Report Options:



Notes:

1 If no options are specified, the TCP TR overall summary report is displayed.

2 Not valid when the -I, -N, -A -S, or -F -S options are specified.

3 Valid only when -A/-C/-F/-T/-U is specified.

4 Valid only when -A/-C/-F/-N/-T/-U is specified.

5 Valid only when -T is specified.

6 Valid only when -A/-F/-T/-U is specified.

7 Valid only when -A/-C/-F/-N/-U is specified.

8 Valid only when -T and -S is specified.

9 Valid only when -A/-T is specified.

10 Not valid with -S or -I.

11 Valid only when -F is specified.

Parameters

-? Displays the help information.

log_file_name
Name of the input file to be analyzed. (The logfile of TRMD.) You must enter a *log_file_name*.

-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.

-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 *z/OS Communications Server: IP Configuration Reference*.

-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/-T/-U option is used. Not valid when the -I, -N, -A -S, or -F -S options are specified.

-A Displays the attack summary.

-C Displays the connection summary.

-F Displays the flood summary.

-I Displays the IDS Overall Summary Report.

-N Displays the scan summary.

-T Displays the TCP TR summary.

-U Displays the UDP TR summary.

-D Displays detailed information. Valid only when the -A/-C/-F/-N/-T/-U option is used (for example, -CD or -C -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.

-h *ip_address*
Displays information about that particular IP address. Valid only when the -A/-C/-F/-N/-U option is used.

-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 with the -A/-T options.

-t *ip_address*
Specifies that information is to be gathered about the destination *ip_address*. Valid only with the -A/-T options.

-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.

Examples

- **Summary Report**

  This is the overall TCP TR summary report. It is displayed when no report options are provided on the trmdstat command invocation.

```
>trmdstat /tmp/tstlog.log
  trmdstat for z/OS CS V1R9          Thu Mar  1 10:16:31 2001

  Stack Name           : ALL
  Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
  Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
  TRM Records Scanned  : 79
  Port Range           : ALL

  Traffic Regulation - TCP
  ------------------------------------------------
1   Connections would have been refused :          3
2   Connections refused                 :          4

3   Constrained entry logged            :          1
4   Constrained exit logged             :          1
5   Constrained entry                   :          1
6   Constrained exit                    :          1

7   QOS exceptions logged               :          2
8   QOS exceptions made                 :          2


  5388 TCP    messages lost at 08/21/2000 09:34:31.03

  TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the summary report.

1 Specifies the number of connections that would have been refused if policy action LIMIT had been specified in the TR policy. This count will indicate the total number of EZZ9319I messages present in the log.

2 Indicates the number of connections refused by the system. This count indicates the total number of EZZ9324I messages present in the log.

3 Specifies the number of times TCP 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.

4 Specifies the number of times TCP 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.

5 Specifies the number of times TCP entered a constrained state. This count indicates the total number of EZZ9321I messages present in the log.

6 Specifies the number of times TCP exited a constrained state. This count indicates the total number of EZZ9323I messages present in the log.

7 Specifies the number of times a QoS exception was made. This count will indicate the total number of EZZ9317I messages present in the log.

8 Specifies the number of times a QoS exception was logged. If policy action LIMIT had been specified in the TR policy, the connection would have been refused. This count will indicate the total number of EZZ9318I messages present in the log.

- **IDS Overall Summary Report**

  This report will be displayed when the -I option is specified with the trmdstat command. It will display the summary of all the IDS information (TCP TR, UDP TR, SCAN, ATTACK and FLOOD) present in the log. Using this report, the user will be able 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 V1R9         Wed Mar 13 15:51:45 2002

Log Time Interval    : Jan  9 12:16:24  - Jan  9 12:20:54
Stack Time Interval  : Jan  9 16:09:06  - Jan  9 17:20:36
TRM Records Scanned  : 3307
Port Range           : ALL

Traffic Regulation - TCP
------------------------------------------------
1   Connections would have been refused :         0
2   Connections refused                 :         0

3   Constrained entry logged            :         0
4   Constrained exit logged             :         0
5   Constrained entry                   :         1
6   Constrained exit                    :         1

7   QOS exceptions logged               :         0
8   QOS exceptions made                 :         0

Traffic Regulation - UDP
------------------------------------------------
9   Constrained entry logged            :         0
10  Constrained exit logged             :         0
11  Constrained entry                   :         0
12  Constrained exit                    :         0


SCAN Detection
------------------------------------------------
13  Threshold exceeded                  :         0
14  Detection delayed                   :         0
15  Storage constrained entry           :         0
16  Storage constrained exit            :         0

ATTACK Detection
------------------------------------------------
17  Packet would have been discarded    :         0
18  Packet discarded                    :       593
19  Accept queue expanded               :         0

FLOOD Detection
------------------------------------------------
20  SYN flood start                     :         0
21  SYN flood end                       :         0
22  Interface flood start               :         0
23  Interface flood end                 :         0

24  440 ATTACK messages lost at 01/09/2002 16:08:26.49
```

  The following describes the areas of the IDS summary report.

  1 Specifies the number of connections that would have been refused if policy action LIMIT had been specified in the TR policy. This count will indicate the total number of EZZ9319I messages present in the log.

  2 Indicates the number of connections refused by the system. This count indicates the total number of EZZ9324I messages present in the log.

3 Specifies the number of times TCP 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.

4 Specifies the number of times TCP 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.

5 Specifies the number of times TCP entered a constrained state. This count indicates the total number of EZZ9321I messages present in the log.

6 Specifies the number of times TCP exited a constrained state. This count indicates the total number of EZZ9323I messages present in the log.

7 Specifies the number of times a QoS exception was made. This count will indicate the total number of EZZ9317II messages present in the log.

8 Specifies the number of times a QoS exception was logged. If policy action LIMIT had been specified in the TR policy, the connection would have been refused. This count will indicate the total number of EZZ9318I messages present in the log.

9 Specifies the number of times UDP 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.

10 Specifies the number of times UDP 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.

11 Specifies the number of times UDP entered a constrained state. This count indicates the total number of EZZ8639I messages present in the log.

12 Specifies the number of times UDP exited a constrained state. This count indicates the total number of EZZ8641I messages present in the log.

13 Specifies the number of scan events detected. This count indicates the total number of EZZ8643I messages present in the log.

14 Specifies the number of scan interval overrun events detected. This count indicates the total number of EZZ8645I messages present in the log.

15 Specifies the number of storage constraint entry was detected. This count indicates the total number of EZZ8646I messages present in the log.

16 Specifies the number of times scan storage constraint exit was detected. This count indicates the total number of EZZ8647I messages present in the log.

17 Specifies the total number of attack packets that would have been discarded if policy action LIMIT had been specified in the Attack policy. This count indicates the total number of EZZ8649I messages present in the log.

18 Specifies the total number of attack packets discarded. This count indicates the total number of EZZ8648I messages present in the log.

19 Specifies the number of accept queue expansions. This count indicates the total number of EZZ8652I messages present in the log.

20 Specifies the number of syn flood starts detected. This count indicates the total number of EZZ8650I messages present in the log.

21 Specifies the number of syn flood ends detected. This count indicates the total number of EZZ8651I messages present in the log.

22 Specifies the number of interface flood starts detected. This count indicates the total number of EZZ8654I messages present in the log.

23 Specifies the number of interface flood ends detected. This count indicates the total number of EZZ8655I messages present in the log.

24 Specifies the number and type of messages lost, with date and time. This data comes from an EZZ9325I message. An EZZ9325I 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.

- **UDP TR Summary Report**

  This report will be displayed when the -U option is specified with the trmdstat command. It will display 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.

```
>trmdstat -U /tmp/tstlog.log
trmdstat for z/OS CS V1R9         Thu Mar  1 11:14:53 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                            UDP  Summary

                               Constrained State          Datagrams
   IP Address      Port   Entered     Exited   Duration   Discarded
----------------- ------ ---------- ---------- ---------- ----------
 05.16.17.18        2001          0          1        100        155
 05.16.17.18        5001          0          2        240        310
 15.16.17.18        5001          2          0          0          0
 25.26.27.28        2001          1          0          0          0


                               Constrained State          Datagrams
   IP Address      Port   Entered     Exited   Duration     Would
                                                          have been
                                                          Discarded
----------------- ------ ---------- ---------- ---------- ----------
 05.16.17.18        1001          1          1        100        141
 05.16.17.18        2001          2          2        220        310
 15.16.17.18        1001          1          1        115        156
 15.16.17.18        3001          1          1        108        105
 15.16.17.18        5001          1          1         90         55

541 UDP    messages lost at 08/21/2000 14:32:33.18


TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the UDP summary report.

IP Address
Specifies the bound IP address.

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 constrained duration in seconds.

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.

messages lost
The number of UDP TR messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **UDP TR Detail Report**

  This report will be displayed when both the -U and -D options are specified. It will display the contents of individual UDP records. The information presented in this report is derived from EZZ8638I, EZZ8639I, EZZ8640I, and EZZ8641I syslog messages.

```
>trmdstat -U -D /tmp/tstlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 11:16:45 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                                UDP  Events

IP Address : 05.16.17.18
    Date and Time       Port   Type   Duration   Discarded  Qsize  Correlator
   ---------------     ------  ---   ----------  ---------- -----  ----------
8/21/2000 14:33:9.52     5001   E                             VL         99221
8/21/2000 14:33:9.53     5001   X            100         155  VL         99227
8/21/2000 14:33:9.53     2001   X            100         155  VS         99228

IP Address : 15.16.17.18
    Date and Time       Port   Type  Duration     Would    Qsize  Correlator
```

```
                                                 have been
                                                 Discarded
   ---------------     ------  ---   ----------  ---------- -----  ----------
8/21/2000 14:31:9.50     1001   E                             VS       87100
8/21/2000 14:32:9.52     3001   E                              L       87101
8/21/2000 14:32:9.53     1001   X           100         155   VS       87225
8/21/2000 14:33:9.55     3001   X           100         155    L       87232

541 UDP    messages lost at 08/21/2000 14:32:33.18

TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the UDP detail report.

IP Address
Specifies the bound IP address.

Date and Time
Specifies the date and time.

Port Specifies the port number.

Type Specifies the entry to or exit from constrained state.

- **E** for enter
- **X** for exit

Duration
Specifies the duration of constrained state in seconds. Present only on EXIT records.

Datagram Disposition
Specifies the datagram disposition. Present only on EXIT records.

- Number of datagrams **Discarded**
- Number of datagrams that **Would have been Discarded** if policy action LIMIT had been specified

Qsize Specifies the qlimit specified on the policy

- **VS** for very small
- **S** for small
- **L** for large
- **VL** for very large

Correlator
Specifies the trace correlator.

messages lost
The number of UDP TR messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **UDP TR Statistics Report**

```
>trmdstat -U -S /tmp/statlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 11:10:24 2001

Stack Name           : ALL
Log Time Interval    : Jan  9 10:54:17  - Jan  9 10:54:25
```

```
Stack Time Interval  : Jan  9 15:47:00  - Jan  9 15:53:49
TRM Records Scanned  : 28
Port Range           : ALL

                               UDP  Statistics


IP Address : 127.0.0.1
    Date and Time       Port    Datagrams Received  Datagrams Discarded  Dgs  Peak
---------------------- ------ -------------------- -------------------- ----- -----
01/09/2001 15:43:00.11   8000             12345670                 1230         111
                                 Bytes Received      Bytes Discarded    Bytes Peak
                              -------------------- -------------------- ----- -----
                                          12345671                 1231        1111
                                     Duration           Constraints    Qsize  Action
                                    ----------           ----------    -----  -------
                                            10                   50      VS   NOLIMIT
    Date and Time       Port    Datagrams Received  Datagrams Discarded  Dgs  Peak
---------------------- ------ -------------------- -------------------- ----- -----
01/09/2001 15:44:01.86   8001             22222220                 2220         222
                                 Bytes Received      Bytes Discarded    Bytes Peak
                              -------------------- -------------------- ----- -----
                                          22222222                 2222        2222
                                     Duration           Constraints    Qsize  Action
                                    ----------           ----------    -----  -------
                                            20                   51      VS   NOLIMIT

IP Address : 127.0.0.2
    Date and Time       Port    Datagrams Received  Datagrams Discarded  Dgs  Peak
---------------------- ------ -------------------- -------------------- ----- -----
01/09/2001 15:45:07.48   8004             55555550                 5550         555
                                 Bytes Received      Bytes Discarded    Bytes Peak
                              -------------------- -------------------- ----- -----
                                          55555555                 5555        5555
                                     Duration           Constraints    Qsize  Action
                                    ----------           ----------    -----  -------
                                            50                   55      VS   LIMIT

110 UDP    messages lost at 01/09/2001 15:54:49.70

TRMD Started                 : Jan  9 10:53:42
TRMD Ended                   : Jan  9 11:05:14
```

The following describes the areas of the UDP statistics report.

IP Address
Specifies the bound IP address.

Date and Time
Specifies the date and time in the message when the statistics were logged.

Port Specifies the 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.

Dgs 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 qlimit specified on the policy.

- **VS** for very small
- **S** for small
- **L** for large
- **VL** for 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.

messages lost

The number of UDP TR statistics messages lost with date and time.

This data comes from an EZZ9326I message. An 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.

- **SCAN Summary Report**

  This report will be displayed when the -N option is specified on the trmdstat command. It will display the summary of scan events. The information presented in this report is derived from EZZ8643I type syslog messages.

```
>trmdstat -N /tmp/tstlog.log
trmdstat for z/OS CS V1R9         Thu Mar  1 10:31:59 2001

Stack Name          : ALL
Log Time Interval   : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned : 79
Port Range          : ALL
```

```
                             SCAN  Summary

  IP Address          Scans                 Suspicion Level
                   Fast      Slow       Very      Possibly     Normal
--------------- ---------- ---------- ---------- ---------- ----------
11.12.13.14              2          2         20         20         20
22.33.44.55              2          0        200        400        600

341 SCAN   messages lost at 08/21/2000 14:32:33.18

TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the Scan summary report.

IP Address
Specifies the bound IP address.

Fast Scan
Specifies the number of fast scans detected.

Slow Scan
Specifies the number of slow scans detected.

Suspicion Level
Specifies the number of packets at each suspicion level that contributed to the scan detection. 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 counts.

messages lost
The number of Scan messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **SCAN Detail Report**

  This report will be displayed when both the -N and -D options are specified on the trmdstat command. It will display the contents of individual scan event records. The information presented in this report is derived from EZZ8643I type syslog messages.

```
>trmdstat -N -D /tmp/tstlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 10:33:04 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                                    SCAN  Events

    Date and Time        IP Address            Suspicion Level
Type Correlator
                                          Very      Possibly   Normal

---------------------- --------------- ---------- ---------- ----------
 --- ----------
8/21/2000 14:32:9.53   11.12.13.14              5          5          5
  S      47331
```

```
8/21/2000 14:32:9.54   22.33.44.55            10         15          0
  F      97338

341 SCAN   messages lost at 08/21/2000 14:32:33.18

TRMD Started                 : Aug 21 08:32:09
```

The following 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.

IP Address
: Specifies the IP address of the source host that triggered the scan detection.

Suspicion Level
: Specifies the number of packets at each suspicion level that contributed to the scan detection. When 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 counts.

Type
: Specifies the scan type.
  - **F** for Fast
  - **S** for slow

Correlator
: Specifies the trace correlator.

messages lost
: The number of Scan messages lost with date and time.

  This data comes from an EZZ9325I message. An EZZ9325I 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.

- **ATTACK Summary Report**

  This report will be displayed when the -A option is specified with the trmdstat command. It will display the summary of all attack events. The information presented in this report is derived from EZZ8648I and EZZ8649I types of syslog messages.

```
>trmdstat -A /tmp/tstlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 10:21:15 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                                        ATTACK  Summary

                                      Datagrams Discarded

Source: 31.32.33.34      Destination: 41.42.43.44
                                            Attacks
  Dst Port   Malf       ORaw       IPFr       ICMP       IPop       Prto
   Perp       NoId
   ------ ---------- ---------- ---------- ---------- ---------- ----------
```

```
---------- ----------
    12001          2          0          0          0          0          0
        1          0
    13001          0          0          0          0          0          0
        1          0


                               Datagrams would have been Discarded

Source: 31.32.33.34      Destination: 41.42.43.44
                                            Attacks
  Dst Port   Malf       ORaw       IPFr       ICMP       IPop       Prto
  Perp       NoId
   ------ ---------- ---------- ---------- ---------- ---------- ----------
---------- ----------
    13001          0         11          0          0          0          0
         0          0

641 ATTACK messages lost at 08/21/2000 14:36:33.18

TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the ATTACK summary report.

Source
Specifies the source IP address.

Destination
Specifies the destination IP address.

Dst Port
Specifies the destination port number.

Malf Specifies the number of malformed packet attacks detected.

ORaw Specifies the number of outbound raw packet attacks detected.

IPFr Specifies the number of IP fragment packet attacks detected.

ICMP Specifies the number of ICMP Redirect packet attacks detected.

IPop Specifies the number of restricted IP option packet attacks detected.

Prto Specifies the number of restricted IP protocol packet attacks detected.

Perp Specifies the number of perpetual echo packet attacks detected.

NoId Specifies the number of EZZ8648I or EZZ8649I messages received with an unknown attack type.

Discarded
Specifies the number of packets discarded.

Would have been Discarded
Specifies the number of packets that would have been discarded if the policy action LIMIT had been specified.

messages lost
The number of Attack messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **ATTACK Detail Report**

  This report will be displayed when both the -A and -D options are specified on the trmdstat command. It will display the contents of attack event records. The information presented in this report is derived from EZZ8648I and EZZ8649I types of syslog messages.

```
>trmdstat -A -D /tmp/tstlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 10:23:53 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 09:35:09
Stack Time Interval  : Aug 21 14:31:31  - Aug 21 14:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                                     ATTACK  Events

                                    Packets Discarded
Attack      Date and Time       Dst IpAddr     Src IpAddr     Dst Port
Src Port Correlator ProbeID
------ --------------------- --------------- --------------- --------
-------- ---------- --------
 Perp  8/21/2000 14:32:9.53   41.42.43.44     31.32.33.34       12001
  10001        9341 04080001
  Malf  8/21/2000 14:32:9.53   51.52.53.54     41.42.43.44       11001
  10001        8334 04010009
 ORAW  8/21/2000 14:32:9.53   51.52.53.54     41.42.43.44       11001
  10001        8335 04020003
 IPFr  8/21/2000 14:32:9.53   51.52.53.54     41.42.43.44       11001
  10001        8336 04030001


                            Packets would have been Discarded
Attack      Date and Time       Dst IpAddr     Src IpAddr     Dst Port
Src Port Correlator ProbeID
------ --------------------- --------------- --------------- --------
-------- ---------- --------
 ORAW  8/21/2000 14:32:9.54   41.42.43.44     31.32.33.34       13001
  11001       87999 04020002

641 ATTACK messages lost at 08/21/2000 14:36:33.18

TRMD Started                 : Aug 21 08:32:09
```

The following describes the areas of the ATTACK detail report.

Attack Specifies the attack type. The values that can be displayed are:

Malf	Malformed Packet
ORaw	OutBound Raw
IPFr	IP Fragment
ICMP	ICMP Redirect
IPop	IP Options
Perp	PerpEcho
PRTO	IP Protocol error
Flod	Flood

NoID Not identified

Date and Time
Specifies the date and time.

Dst IpAddr
Specifies the destination IP address.

Src IpAddr
Specifies the source IP address.

Dst Port
Specifies the destination port.

Src Port
Specifies the source port.

Correlator
Specifies the trace correlator.

ProbeID
Specifies the IDS probeID that generated this event.

messages lost
The number of Attack messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **ATTACK Statistics Report**

  This report will be displayed when both the -A and -S options are specified on the trmdstat command. It will display 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 of records. For the FLOD type, the attacks number will represent the total number of SYN flood and Interface flood starts detected during the interval.

```
>trmdstat -A -S /tmp/statlog.log
trmdstat for Z/OS CS V1R9         Tue Jan 16 13:13:30 2001

Log Time Interval    : Jan  9 10:54:15  - Jan  9 10:54:16
Stack Time Interval  : Jan  9 15:42:53  - Jan  9 15:45:58
TRM Records Scanned  : 27
Port Range           : ALL

                         ATTACK TR Statistics

Attack          Date and Time             Attacks          Action
------        ----------------------    ----------        -------
 Malf         01/09/2001 15:42:53.20         11111        LIMIT
 IPFr         01/09/2001 15:42:53.20         22222        LIMIT
 ORAW         01/09/2001 15:43:54.84         33333        LIMIT
 IPFr         01/09/2001 15:43:54.84         44444        LIMIT
 ICMP         01/09/2001 15:44:56.52         55555        LIMIT
 IPOP         01/09/2001 15:44:56.52         66666        NOLIMIT
 Perp         01/09/2001 15:45:58.17         77777        NOLIMIT
 186 ATTACK messages lost at 01/09/2001 21:51:14.75

TRMD Started                : Jan  9 10:53:42
```

The following describes the areas of the ATTACK statistics report.

Attack Indicates the ATTACK type causing the packet to be discarded if the statistics record indicates LIMIT or would have been discarded if the statistics record indicates NOLIMIT. The values that can be displayed are:

Malf	Malformed Packet
ORaw	OutBound Raw
IPFr	IP Fragment
ICMP	ICMP Redirect
IPop	IP Options
Perp	PerpEcho
PRTO	IP Protocol error
Flod	Flood
NoID	Not identified

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 specified on the policy ibm-idsTypeActions statement. LIMIT indicates that policy action LIMIT was specified. NOLIMIT indicates tht policy action LIMIT was not specified.

messages lost
The number of Attack Statistics messages lost with date and time.

This data comes from an EZZ9326I message. An 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.

- **FLOOD Summary Report**

  This report will be displayed when the -F option is specified with the trmdstat command. It will display the summary of all flood events. The information presented in this report is derived from EZZ8650I, EZZ8651I, EZZ8654I, and EZZ8655I types of syslog messages. Summary data related to Syn Floods and Interface Floods will be shown in separate sections of the report.

```
> trmdstat -F /u/user1/tstlog.log
trmdstat for z/OS CS V1R9         Mon Feb 24 13:29:45 2003

Stack Name           : ALL
Log Time Interval    : Feb 19 11:13:01  - Feb 19 11:22:01
Stack Time Interval  : Feb 19 16:12:51  - Feb 19 16:21:43
TRM Records Scanned  : 9
```

```
Port Range           : ALL

                       SYN FLOOD  Summary

  IP Address       Port   SYN Flood   SYN Flood   SYN Flood
                            Start        End       Duration
---------------   ------  ----------  ----------  ----------
9.42.104.38          215           1           1         532

                       Interface Flood Summary

                            IFC Flood   IFC Flood   IFC Flood
Interface Name                Start        End       Duration
----------------            ----------  ----------  ----------
ETH1                                 1           1         365
```

The following describes the areas of the SYN FLOOD summary report.

IP Address
Specifies the bound IP address.

Port Specifies the bound port number.

SYN Flood Start
Specifies the number of SYN Flood attack starts.

SYN Flood End
Specifies the number of SYN Flood attack ends.

SYN Flood Duration
Specifies the SYN Flood durations in seconds.

The following describes the areas of the Interface FLOOD summary report.

Interface Name
Specifies the interface (or link) name of the interface for which an interface flood was detected.

IFC Flood Start
Specifies the number of Interface floods starts detected.

IFC Flood End
Specifies the number of Interface floods ends detected.

IFC Flood Duration
Specifies the accumulated duration of the interface floods that have ended in seconds. Duration is non-zero only if the interface has experienced at least one flood that has ended.

- **FLOOD Detail Report**

  This report will be displayed when both the -F and -D options are specified with the trmdstat command. It will display the contents of flood event records. The information presented in this report is derived from EZZ8650I, EZZ8651I, EZZ8654I, EZZ8655I, and EZZ8656I types of syslog messages.

  Data related to Syn Floods and Interface Floods will be shown in separate sections of the report. 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 /u/user1/tstlog.log
trmdstat for z/OS CS V1R9          Mon Feb 24 13:30:41 2003

Stack Name           : ALL
```

```
Log Time Interval    : Feb 19 11:13:01  - Feb 19 11:22:01
Stack Time Interval  : Feb 19 16:12:51  - Feb 19 16:21:43
TRM Records Scanned  : 9
Port Range           : ALL

                                                   SYN FLOOD  Events

   Date and Time        IP Address      Port  Type  SYNsRecvd   FirstAck  SYNsDiscd  SYNsTimeO   Duration  Correlator
---------------------- --------------- ------  ---  ---------- ---------- ---------- ---------- ---------- ----------
02/19/2003 16:12:51.36 9.42.104.38       215   E                                                                  20
02/19/2003 16:21:43.40 9.42.104.38       215   X        10548          0      10548        257        532         20

                                                   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
02/19/2003 16:13:35.55     ETH1              E                  1000         21
           9.0.0.1                                                95   04070010
           9.42.104.38
02/19/2003 16:18:59.90     ETH1              C        304      10643         21         6     Queue  00D06355D820       6     Queue
       1   9.42.104.1                                             96   04070011        86        86            90      95        95
           9.42.104.38
02/19/2003 16:20:05.80     ETH1              X        365      10645         21         6     Queue  00D06355D820       6     Queue
       3   9.42.104.1                                             96   04070014        86        86            90      95        95
           9.42.104.38
```

The following describes the areas of the SYN FLOOD detail report.

Date and Time
Specifies the date and time.

IP Address
Specifies the bound IP address.

Port Specifies the port number.

Type Specifies the entry to or exit from constrained state.

- **E** for enter
- **X** for exit

SYNsRecvd
Handshakes started during syn flood. Present only on EXIT records.

FirstAck
Handshakes completed during syn flood. Present only on EXIT records.

SYNsDiscd
SYNs randomly discarded during syn flood. Present only on EXIT records.

SYNsTimeO
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 interface (or link) name experiencing the interface flood condition.

Type Specifies the entry to, or exit from flood state, or a continuing 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 discarded packets that triggered the interface flood detection. On interface flood exit or continuation, this is the percentage of discarded packets detected 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 will be 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.

NETACC

Packet dropped due to NetAccess checks.

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.

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 the microcode level that supports providing the source MAC address of the prior hop. It is not applicable for other devices.

SrcMAC/Percent

SrcMAC
Source MAC of the prior hop seen most frequently in the discarded packets. The value N/A will appear 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.

For interface floods, the duration, most frequent and last source IP information is only available on the flood exit or flood continuing (Type X or C) log records.

If IP address filtering (-h) is requested, the interface flood records are filtered using the destination address. The Interface Flood Events report width is 132 characters. If you are displaying or printing this report, use an output device that can accomodate this width.

- **FLOOD Statistics 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 of records. An overall flood statistics log record, EZZ8653I with attack type Flod, 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 ibm-idsIfcFloodPercentage and ibm-idsIfcFloodMinDiscard, 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.

```
>trmdstat -FS   /tmp/syslog.info
trmdstat for z/OS CS V1R9          Tue Nov 12 17:13:44 2002

Stack Name           : ALL
Log Time Interval    : Nov  4 15:22:32  - Nov 12 18:18:16
Stack Time Interval  : Nov 04 15:22:18  - Nov 12 18:18:05
TRM Records Scanned  : 266
Port Range           : ALL

                  Overall FLOOD Statistics
  Date and Time         Flood Count
----------------------   ----------
11/04/2002 15:22:18.73            1
11/04/2002 20:44:50.65            1
11/04/2002 21:28:11.43            1
11/04/2002 21:43:32.25            1


                  Interface FLOOD Detailed Statistics
  Date and Time        Interface         -----Discard-----     Attacks
                                           Count       Pct
---------------------- ----------------  ----------    ---    ----------
11/04/2002 15:22:18.73 QDIO1                    278     21             1
11/04/2002 20:44:50.65 TR1                      539     18             1
11/04/2002 21:28:11.43 TR1                      238     72             1
11/04/2002 21:43:32.25 TR1                      318     76             1
```

The following 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
Interface (or link) name for which the data is reported.

Discard Count
Number of inbound packets discarded or not processed during the statistics interval.

Discard Pct
Percentage of discarded packets detected on the interface during the statistics interval.

Attacks
Number of Interface floods entries detected on the interface during the statistics interval.

- **TCP TR Summary Report**

  This report will be displayed when the -T option is specified with the trmdstat command. It will display 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.

```
>trmdstat -T /tmp/tstlog.log
trmdstat for z/OS CS V1R9          Thu Mar  1 10:57:22 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 15:35:09
Stack Time Interval  : Aug 21 09:31:31  - Aug 21 15:34:33
TRM Records Scanned  : 79
Port Range           : ALL

                               TCP  Summary


 Local Host: 00.01.02.03      Source Host: ALL
                Constrained States                      Connections
  Port               Limited                Excp          Refused
           Enter      Exit     Duration     QOS        Appl       Host
 ------ ---------- ---------- ---------- ---------- ---------- ----------
   3001          1          1        123          0          3          0
   7001          0          0          0          2          0          0
   8001          0          0          0          0          0          1

 Local Host: 20.21.22.23      Source Host: ALL
                Constrained States                      Connections
  Port               Logged                 Excp   Would have been Refused
           Enter      Exit     Duration     QOS        Appl       Host
 ------ ---------- ---------- ---------- ---------- ---------- ----------
   2001          1          1          7          0          2          0

5388 TCP messages lost at 08/21/2000 09:34:31.03
```

The following describes the areas of the TR TCP Summary Report:

Local Host
If the policy action specified ibm-idsTRtcpLimitScope:PORT this will always be 255.255.255.255. If the policy action specified ibm-idsTRtcpLimitScope:PORT_INSTANCE this will be the IP address bound to by the local listener applications. A value of 0.0.0.0 indicates the application bound to InAddrAny.

Source Host
Indicates the source IP address specified on filter -s or ALL if none was specified.

Limited | Logged

For each Local Host-Source Host pair 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.

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 the ibm-idsTRtcpPercentage allowed while the port was not constrained.

Connections Refused Appl

The number of connections refused because the ibm-idsTRtcpTotalConnections limit was exceeded.

Connections Refused Host

The number of connections refused because the number of connections requested from a single host exceeded the ibm-idsTRtcpPercentage of remaining available connections.

Connections Would Have Been Refused Appl

The number of connections that would have been refused because the ibm-idsTRtcpTotalConnections limit was exceeded if policy action LIMIT was 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 ibm-idsTRtcpPercentage of remaining available connections if policy action LIMIT was specified.

messages lost

The number of TCP TR messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **TCP TR Detail Report**

  This report will be displayed when both the -T and -D options are specified with the trmdstat command. It will display the contents of individual TCP TR records. The information presented in this report is derived from EZZ9317I, EZZ9318I, EZZ9319I, EZZ9320I, EZZ9321I, EZZ9322I, EZZ9323I, and EZZ9324I types of syslog messages.

```
>trmdstat -T -D /tmp/tstlog.log
trmdstat for z/OS CS V1R9         Thu Mar  1 11:00:17 2001

Stack Name           : ALL
Log Time Interval    : Aug 21 09:32:09  - Aug 21 15:35:09
Stack Time Interval  : Aug 21 09:31:31  - Aug 21 15:34:33
TRM Records Scanned  : 79
Port Range           : ALL
```

```
                                                       TCP  Events

                                                      Events Limited

 Local Host: 00.01.02.03      Source Host: ALL
     Date and Time       Port    Source Host   Rec Cns     Connections
          Policy          Correlator ProbeID
                                               Typ Typ  Current   Available
 Total Conn Pct Qos Limit
    ---------------     ------ --------------- --- --- ---------- ----------
 ---------- --- ---------- ---------- --------
 8/21/2000 10:14:06.12    3001 10.10.10.240     S   E          91         9
        100   5         0         15 01004400
 8/21/2000 10:42:17.81    3001 10.10.10.151     C             100         0
        100   5         0         15 01004048
 8/21/2000 11:02:41.53    7001 10.11.12.13      Q             170        30
        200  10         5         33 01004014
 8/21/2000 11:12:39.25    3001 10.10.10.8       C             100         0
        100   5         0         15 01004048
 8/21/2000 11:32:09.54    8001 11.12.13.14      C             411       589
       1000  25         0         34 01004044
 8/21/2000 11:41:16.57    3001 10.10.10.93      C             100         0
        100   5         0         15 01004048
 8/21/2000 12:17:44.15    3001 10.10.10.240     S   X          87        13
        100   5         0         15 01004400
 8/21/2000 14:32:27.55    7001 10.11.12.13      Q             176        24
        200  10         5         33 01004014

                                                      Events Logged

 Local Host: 20.21.22.23      Source Host: ALL
     Date and Time       Port    Source Host   Rec Cns     Connections
          Policy          Correlator ProbeID
                                               Typ Typ  Current   Available
 Total Conn Pct Qos Limit
    ---------------     ------ --------------- --- --- ---------- ----------
 ---------- --- ---------- ---------- --------
 8/21/2000 14:32:09.55    2001 11.12.13.14      S   E         451        49
        500  75         0         41 01004400
 8/21/2000 14:32:12.61    2001 11.12.13.14      C             500         0
        500  75         0         41 01004800
 8/21/2000 14:37:44.18    2001 11.12.13.95      C             501        -1
        500  75         0         41 01004800
 8/21/2000 14:39:31.16    2001 11.12.13.135     S   X         440        60
        500  75         0         41 01008800

5388 TCP messages lost at 08/21/2000 09:34:31.03
```

The following describes the areas of the TR TCP Detail Report:

Events Limited | Logged

For each Local Host-Source Host pair 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 action specified ibm-idsTRtcpLimitScope:PORT this will always be 255.255.255.255. If the policy action specified ibm-idsTRtcpLimitScope:PORT_INSTANCE this will be the IP address bound to by the local listener applications. A value of 0.0.0.0 indicates the application bound to InAddrAny.

Source Host:
Indicates the source IP address specified on filter -s or ALL if none was specified.

Date and Time
The stack date and time the event occurred.

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

S Port entered or exited constraint events

Cns Typ
Constraint event type.

E Entered

X Exited

Connections Current
The current number of connections, at the time of this event, to this port, made 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 ibm-idsTRtcpTotalConnections limit for this port.

Policy Pct.
The ibm-idsTRtcpPercentage limit for this port.

Policy QoS Limit
The ibm-MaxConnections 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 lost
The number of TCP TR messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **TCP Extended Summary Report**

  This report will be displayed when both the -T and -E options are specified with the trmdstat command. It will display an extended summary of all TCP traffic regulation events. For each port a seperate 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.

```
 > trmdstat -T -E /tmp/tstlog.log
 trmdstat for z/OS CS V1R9         Thu Mar  1 11:03:32 2001

 Stack Name           : ALL
 Log Time Interval    : Aug 21 09:32:09  - Aug 21 15:35:09
 Stack Time Interval  : Aug 21 09:31:31  - Aug 21 15:34:33
 TRM Records Scanned  : 79
 Port Range           : ALL

                                   TCP  Extended  Summary


 Local Host: 00.01.02.03      Source Host: ALL
                              Constrained States
    Connections
  Port       Host                  Limited                Excp
      Refused
                          Enter      Exit     Duration     QOS
   Appl       Host
 ------ --------------- ---------- ---------- ---------- ----------
 ---------- ----------
   3001 11.12.13.14              1          1        123          0
          3          0
   7001 10.11.12.13              0          0          0          2
          0          0
   8001 11.12.13.14              0          0          0          0
          0          1

 Local Host: 20.21.22.23      Source Host: ALL
                              Constrained States
    Connections
  Port       Host                  Logged                 Excp  Wo
uld have been Refused
                          Enter      Exit     Duration     QOS
   Appl       Host
 ------ --------------- ---------- ---------- ---------- ----------
 ---------- ----------
   2001 11.12.13.14              1          1          7          0
          2          0

5388 TCP messages lost at 08/21/2000 09:34:31.03
```

The following describes the areas of the TR TCP Extended Summary Report:

Local Host
: If the policy action specified ibm-idsTRtcpLimitScope:PORT this will always be 255.255.255.255. If the policy action specified ibm-idsTRtcpLimitScope:PORT_INSTANCE this will be the IP address bound to by the local listener applications. A value of 0.0.0.0 indicates the application bound to InAddrAny.

Source Host:
: Indicates the source IP address specified on filter -s or ALL if none was specified.

Limited | Logged
: For each Local Host-Source Host pair 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.

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 the ibm-idsTRtcpPercentage allowed while the port was not constrained.

Connections Refused Appl
The number of connections refused because the ibm-idsTRtcpTotalConnections limit was exceeded.

Connections Refused Host
The number of connections refused because the number of connections requested from a single host exceeded the ibm-idsTRtcpPercentage of remaining available connections.

Connections Would Have Been Refused Appl
The number of connections that would have been refused because the ibm-idsTRtcpTotalConnections limit was exceeded and a policy action of LIMIT was 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 ibm-idsTRtcpPercentage of remaining available connections if a policy action of LIMIT was specified.

messages lost
The number of TCP TR messages lost with date and time.

This data comes from an EZZ9325I message. An EZZ9325I 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.

- **TR TCP Connection Detail Report**

```
trmdstat -C -D /tmp/tstlog.log
trmdstat for z/OS CS V1R9         Tue Jul 22 13:48:46 2003

Stack Name           : ALL
Log Time Interval    : Jul 18 13:01:02  - Jul 18 13:01:32
Stack Time Interval  : Jul 18 17:00:48  - Jul 18 17:01:05
TRM Records Scanned  : 7
Port Range           : ALL

Port Number      Connections Refused     IP Address
-----------      -------------------     ----------
No records to display

Port Number      Connections Would Have Been Refused        IP Address
-----------      -----------------------------------        ----------
      23                   1
                                1                           9.42.105.120
```

```
       2000                    4
                                     1                         9.42.105.120
                                     2                         9.42.105.122
                                     1                         9.42.105.125
```

The following describes the areas of the TR TCP Connection Detail Report:

This report first shows the total number of ’Connections Refused’ or 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 ibm-idsTypeActions Limit. Data under the ’Connections Would Have Been Refused’ heading are connections that exceeded policy limits but were not rejected because policy did not specified ibm-idsTypeActions Limit.

Port Number
Indicates the port number to which the connection was destined.

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 ibm-idsTypeActions 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 ibm-idsTypeActions Limit was specified in the policy.

IP Address
Indicates the source IP address.

- **TCP TR Statistics Report**

  This report will be displayed when both the -T and -S options are specified on the trmdstat command. It will display the contents of the TCP traffic regulation statistics records, EZZ9316I.

```
>trmdstat -T -S /tmp/statlog.log
trmdstat for z/OS CS V1R9          Thu May 31 17:00:30 2001

Stack Name           : ALL
Log Time Interval    : May 22 13:51:35  - May 22 14:40:35
Stack Time Interval  : May 22 13:51:01  - May 22 14:40:26
TRM Records Scanned  : 28529
Port Range           : ALL

                                           TCP TR Statistics


Local Host: 0.0.0.0          Peak Host: ALL
    Date and Time       Port        Action         Peak     Requests   Warnings
 QosExcepts Terminates
                                   Peak Host     HostPeak   Current    Duration
  SugLimit  SugPercent
---------------------- ------   --------------- ---------- ---------- ----------
 ---------- ----------
05/22/2001 13:52:02.97  50024       NOLIMIT             24         12         12
          0        12
                                130.11.176.103          24         24         61
         55        77
```

```
Local Host: 0.0.0.0          Peak Host: ALL
    Date and Time       Port        Action         Peak     Requests   Warnings
 QosExcepts Terminates
                                   Peak Host     HostPeak   Current    Duration
  SugLimit  SugPercent
---------------------- ------   --------------- ---------- ---------- ----------
 ---------- ----------
05/22/2001 14:26:51.90  45621       NOLIMIT              10          0         0
          0         10
                                130.11.176.103            9          0         3
         21         81


Local Host: 0.0.0.0          Peak Host: ALL
    Date and Time       Port        Action         Peak     Requests   Warnings
 QosExcepts Terminates
                                   Peak Host     HostPeak   Current    Duration
  SugLimit  SugPercent
---------------------- ------   --------------- ---------- ---------- ----------
 ---------- ----------
05/22/2001 14:22:02.01     80       NOLIMIT           37522         45        45
          0          0
                                130.11.176.103        37522      37522        17
      86300         76

05/22/2001 14:23:03.32     80       NOLIMIT           37762         48        48
          0          0
                                130.11.176.103        37762      37762        17
      86852         76

05/22/2001 14:24:04.70     80       NOLIMIT           37959         77        77
          0          0
                                130.11.176.103        37959      37959        17
      87305         76
```

The following describes the areas of the TR TCP Statistics Report:

Local Host
If the policy action specified ibm-idsTRtcpLimitScope:PORT this will always be 255.255.255.255. If the policy action specified ibm-idsTRtcpLimitScope:PORT_INSTANCE this will be the IP address bound to by the local listener applications. A value of 0.0.0.0 indicates the application bound to InAddrAny.

Peak Host
Indicates the source IP address specified on filter -k or ALL if none was specified.

Date and Time
The stack date and time the statistics were reported.

Port The port bound to by a local listener application.

Action
Indicates whether or not an action of LIMIT was specified in the policy 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 the ibm-idsTRtcpPercentage allowed while the port was not constrained.

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 ibm-idsTRtcpTotalConnections 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 will be 0.

SugPercent
A suggested companion value for ibm-idsTRtcpPercentage. If a policy action of LIMIT was in effect then this value will be 0.

messages lost
The number of TCP TR Statistics messages lost with date and time.

This data comes from an EZZ9326I message. An 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.

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), or QoS exception (EZZ9318I) log 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:
`count` `TCP` *`type`* `messages suppressed at` *`time`* `for local host` *`laddr`* `port`
`lport` `scope` *`rsn`*

The values are:

count The number of log records suppressed during the 5-minute interval.

type A value of either `connection refused` or `QOS exception`.

time The stack time when the first log record in the 5-minute interval was suppressed.

laddr The local IP address.

lport The local listening port.

rsn 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.

The TCP TR suppressed count messages for connections refused can be written for the Summary report, the IDS Overall Summary report and any of the TCP summary or detail reports that are requested with the -T or -C option. The TCP TR suppressed count messages for QoS exceptions can be written for the IDS Overall Summary report and any of the TCP summary or detail reports that are requested with the -T option.

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 desired, 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 attack packets that are discarded (EZZ8648I) or would have been discarded (EZZ8649I) for a particular attack type to 100 log records written in a 5-minute interval. 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. Instead the counts are listed at the end of the requested report in the following format:

```
count ATTACK type messages suppressed at time
```

The values are:

count The number of log records suppressed during the 5-minute interval.

type The attack type. Can be one of the following: Malformed, OutboundRaw, IPFragment , ICMP, IPOPT, IPPROTO, FLOOD, or PerpEcho.

time The stack time when the first log record in the 5-minute interval was suppressed.

The attack-suppressed count messages can be written for the IDS Overall Summary report and any of the Attack summary or detail reports.

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 descriptions of the following:

- 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
- **dnssec** commands (**keygen**, **makekeyset**, **signkey**, and **signzone**) related to DNS security
- **rndc** command to remotely control a name server
- **rndc-confgen** command to generate configuration files for rndc.
- **dnsmigrate** command to convert `named.boot` file syntax into `named.conf` file syntax

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" on page 662
- **onslookup**/**nslookup**, see "Using the z/OS UNIX onslookup/nslookup command" on page 676
- nsupdate, see "Using the z/OS UNIX nsupdate command" on page 693
- DIG (TSO), see "Using the TSO DIG command" on page 704
- **dig** (z/OS UNIX), see "Using the z/OS UNIX dig command" on page 722
- host, see "Using the z/OS UNIX host command" on page 733

The BIND 4 **onslookup** and TSO DIG commands use the z/OS Communications Server provided resolver for all their 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

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

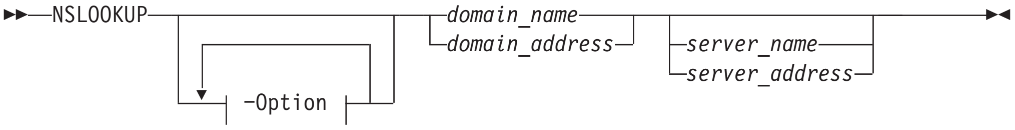
For information about the TCPIP.DATA data set, see the *z/OS Communications Server: IP Configuration Reference*. For information about the NSLOOKUP.ENV options data set and the NSLOOKUP command options, see “NSLOOKUP options” on page 668.

NSLOOKUP—Query a name server in command mode

Purpose

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 668.

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

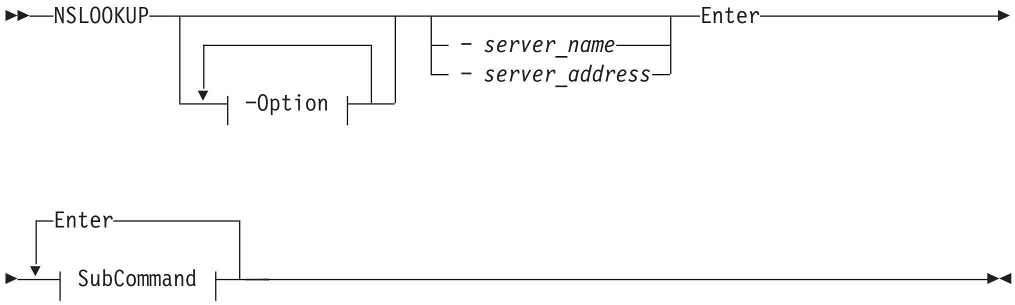
- See “NSLOOKUP options” on page 668 for the complete list and description of NSLOOKUP options.
- See “NSLOOKUP—Issue queries to name servers in interactive mode” on page 664 for the complete list and description of subcommand and query formats.

NSLOOKUP—Issue queries to name servers in interactive mode

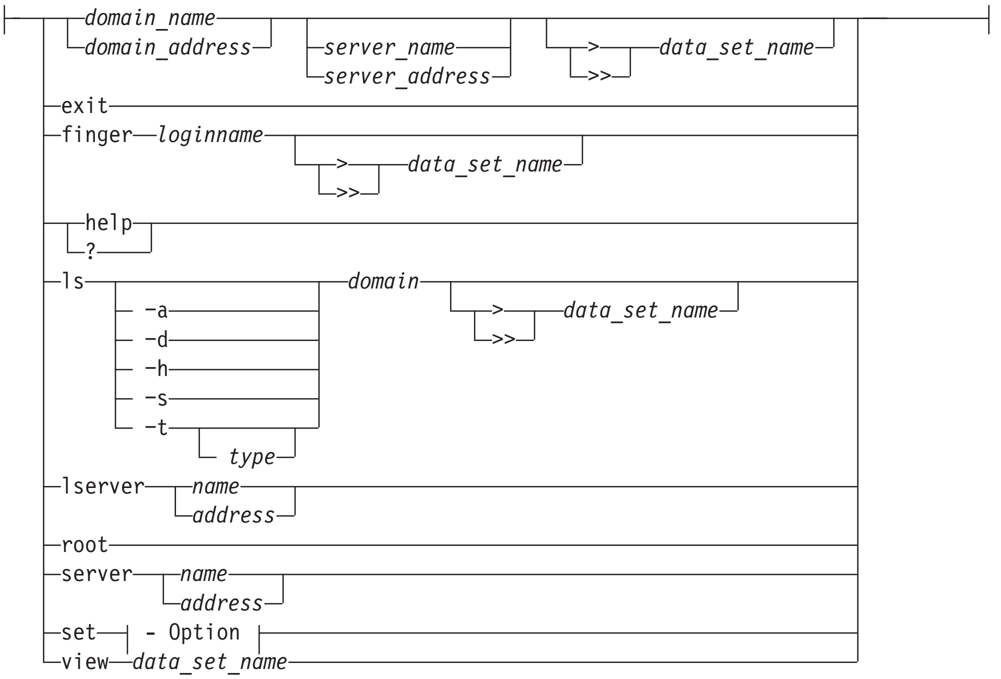
Purpose

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



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 668.

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 668 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 is an example of the contents of the *user_id*.NSLOOKUP.ENV data set:

```
set domain=powers.oz
querytype=HINFO
set norecurse
vc
```

Option:

```
├──┬─all──────────────────────┬──────────────────────────────────────────┤
   │ ┌─nobrackets─┐           │
   ├─┴─brackets───┴───────────┤
   ├─class=class──────────────┤
   │ ┌─nod2─┐                 │
   ├─┴─d2───┴─────────────────┤
   │ ┌─nodebug─┐              │
   ├─┴─debug───┴──────────────┤
   │ ┌─nodefname─┐            │
   ├─┴─defname───┴────────────┤
   ├─domain=name──────────────┤
   │ ┌─noignoretc─┐           │
   ├─┴─ignoretc───┴───────────┤
   │ ┌─53──────────────┐      │
   ├─┴─port──=──port───┴──────┤
   │ ┌─A──────────────────┐   │
   ├─┴─querytype──=──type─┴───┤
   │ ┌─recurse───┐            │
   ├─┴─norecurse─┴────────────┤
   ├─retry=limit──────────────┤
   ├─root=name────────────────┤
   │ ┌─search───┐             │
   ├─┴─nosearch─┴─────────────┤
   │            ┌─/──────┐    │
   │            ▼        │    │
   ├─srchlist=──┴─domain─┴────┤
   ├─timeout=interval─────────┤
   │ ┌─novc─┐                 │
   └─┴─vc───┴─────────────────┘
```

all
Enables you to print the current values of the internal state variables. This option does not alter the internal state of NSLOOKUP.

brackets
Causes output to display with brackets.

nobrackets
Causes output to display using < and > instead of brackets. This option is for terminals that do not support brackets. This is the default.

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.

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. This is the default. 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. Each time a request is resent, the timeout period for the request is twice the timeout period used for the last attempt. 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 unqualified host names when attempting to resolve the host name. Each domain name specified is tried in turn until a match is found.

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 section 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.

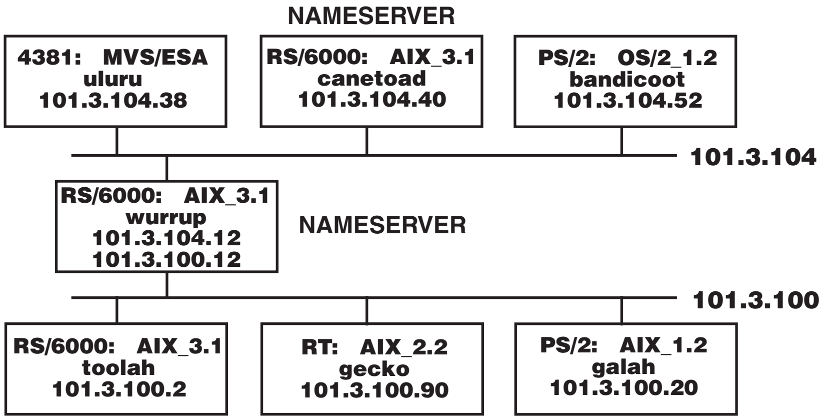


Figure 2. Hierarchical naming tree—A TCP/IP network

The following are examples of 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
   brackets
   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. It also has two versions in z/OS UNIX: v4 and v9, where v4 gives the legacy z/OS UNIX **onslookup** function, and v9 gives the BIND 9 version of **nslookup**. Use the interactive mode to query name servers for information about various hosts and domains or to display a list of hosts (BIND 4) in a domain. Non-interactive mode is used to display just the name and requested information for a host or domain.

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 677 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
```

Notes:

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 will be used.

```
domain   raleigh.ibm.com
nameserver   9.37.34.149
nameserver   9.37.34.7
```

onslookup configuration

The configuration options of v determine the operation and results of your name server queries. The values for **onslookup** options can be specified in more than one location.

- The search order locations and order of priority from which the values for version BIND 4.9.3 **onslookup** options can be specified are:
  1. **onslookup** command options
  2. `.onslookuprc` file in the home directory
  3. Environment variable (LOCALDOMAIN)
  4. TCPIP.DATA statement values (See the *z/OS Communications Server: IP Configuration Guide* for the TCPIP.DATA search order for z/OS UNIX applications.)
- 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`

Values specified as **onslookup** command options have priority over values specified in the `.onslookuprc` file, which have priority over the values specified by the environmental variable, and so on. (This is valid for BIND 4.9.3 **nslookup** only.) For example, the value specified by the *all* option in the **nslookup** command has priority over the value specified by the *all* option in the `.onslookuprc` file. Similarly, 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 661 for detailed descriptions.

See the *z/OS Communications Server: IP Configuration Guide* for detailed information about **onslookup** configuration.

nslookup versions

This section presents similarities and differences for the two separate versions (v4 and v9) 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** v9 which can only be invoked from the z/OS UNIX shell.

The following table shows the validity of start options between TSO, v4, and v9 **nslookup** commands when using the -V (version) start option from the UNIX Systems Services.

Table 14. Start option validity for NSLOOKUP TSO/v4/v9

Start option	TSO NSLOOKUP	Valid with -V=v4	Valid with -V=v9
-all	Yes	Yes	Yes
-brackets	Yes	No	No
-nobrackets	Yes	No	No
-class	Yes	Yes	Yes
-cl (short for -class)	Yes	Yes	Yes
-diffstamp	No	No	No
-d2	Yes	Yes	Yes
-nod2	Yes	Yes	Yes
-debug	Yes	Yes	Yes
-deb (short for -debug)	Yes	Yes	Yes
-nodebug	Yes	Yes	Yes
-nodeb (short for -nodebug)	Yes	Yes	Yes
-defname	Yes	Yes	Yes
-def (short for -defname)	Yes	Yes	Yes
-nodefname	Yes	Yes	Yes
-nodef (short for -nodefname)	Yes	Yes	Yes
-domain=	Yes	Yes	Yes
-do= (short for -domain=)	Yes	Yes	Yes
-help	No	Yes	No
-h	No	No	Yes
-ignoretc	Yes	Yes	No
-ig (short for -ignoretc)	Yes	Yes	No
-noignoretc	Yes	Yes	No
-noig (short for -noignoretc)	Yes	Yes	No
-port=	Yes	Yes	Yes
-po= (short for -port=)	Yes	Yes	Yes
-querytype=	Yes	Yes	Yes
-q= (short for -querytype)	Yes	Yes	No
-type= (short for -querytype=)	Yes	Yes	Yes
-ty= (short for -querytype=)	No	No	Yes
-query= (short for -querytype=	No	No	Yes
-qu= (short for -querytype=)	No	No	Yes
-recurse	Yes	Yes	Yes
-rec (short for -recurse)	Yes	Yes	Yes
-norecurse	Yes	Yes	Yes
-norec (short for -norecurse)	Yes	Yes	Yes
-retry=	Yes	Yes	Yes
-ret= (short for -retry=)	Yes	Yes	Yes

Table 14. Start option validity for NSLOOKUP TSO/v4/v9 (continued)

Start option	TSO NSLOOKUP	Valid with -V=v4	Valid with -V=v9
-root=	Yes	Yes	No
-search	Yes	Yes	Yes
-sea (short for -search)	Yes	Yes	Yes
-nosearch	Yes	Yes	Yes
-nosea (short for -nosearch)	Yes	Yes	Yes
-srchlist=	Yes	Yes	No
-srchl= (short for -searchlist=)	Yes	Yes	No
-timeout=	Yes	Yes	Yes
-t= (short for -timeout=)	Yes	Yes	Yes
-tstamp	No	No	No
-nostamp	No	No	No
-sil	No	No	Yes
-vc	Yes	Yes	Yes
-novc	Yes	Yes	Yes
-V=	No	Yes	Yes

The following table shows the validity of the subcommands between v4 **nslookup** and v9 **nslookup** command when using the -V (version) start option from the UNIX Systems Services.

Table 15. Subcommand validity, NSLOOKUP v4/v9

Subcommand	Valid with -V=v4	Valid with -V=v9
set	Yes, options limited by table above.	Yes, options limited by table above.
server	Yes	Yes
lserver	Yes	Yes
exit	Yes	Yes
help	Yes	No
?	Yes	No
ls	Yes	No
root	Yes	No
view	Yes	No

The following table shows the differences in options supported by TSO NSLOOKUP and v4 **onslookup** commands.

Table 16. Differences between TSO NSLOOKUP and v4 **onslookup**

Option	Support provided by:	
	TSO NSLOOKUP	v4 onslookup
brackets	yes	no
nobrackets	yes	no

onslookup/nslookup (command mode)—Querying a name server in command mode

Purpose

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 Internet 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.

Notes:

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 688.

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 677. For v9 **nslookup** only, this can be a name that resolves to an IPv6 address. If the server exists on an IPv6-only host, the server name or address must be specified, as IPv6 addresses cannot be used in the resolver configuration file.

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 677. For v9 **nslookup** only, this can be an IPv6 address. If the server exists on an IPv6-only host, the server name or address must be specified, as IPv6 addresses cannot be used in the resolver configuration file.

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 688.

onslookup/nslookup (interactive mode)—Issuing multiple queries to name servers

Purpose

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 Internet address of a name server.

Note: The **onslookup help** command works only in the interactive mode.

In interactive mode:

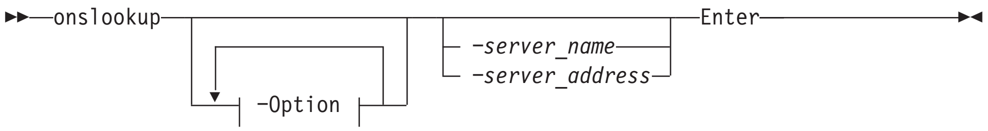
- An initial query is made to the selected name server to verify that the server is accessible.
- All subsequent interactive queries are sent to that server unless you specify another server using the *server* or *lserver* subcommands.
- The command line length must be less than 256 characters.
- To 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 688. See “nslookup versions” on page 677 for listing of valid commands and start options.

Note: Unless specified otherwise, the **onslookup** commands, options, resource records, and keywords are valid with both BIND 4.9.3 and BIND 9.

Format

- Interactive mode



- Interactive commands, subsequent queries

```
►►──onslookup──┬──────────────────────────────────────────────────────┬──────────►
               ├─exit─────────────────────────────────────────────────┤
               │        (1)                                           │
               ├─┬─help─┬─────────────────────────────────────────────┤
               │ └─?────┘                                             │
               ├─host──┬────────┬─────────────────────────────────────┤
               │       └─server─┘                                     │
               │                                                 (1)  │
               ├─ls──┬────────┬─domain──┬──────────────────────┬──────┤
               │     └─ls_opt─┘         └─┬─ > ──┬─file_name───┘      │
               │                          └─ >> ─┘                    │
               ├─lserver──┬─name────┬─────────────────────────────────┤
               │          └─address─┘                                 │
               │      (1)                                             │
               ├─root─────────────────────────────────────────────────┤
               ├─server──┬─name────┬──────────────────────────────────┤
               │         └─address─┘                                  │
               ├─set─keyword──┬──────────┬────────────────────────────┤
               │              └─=─value──┘                            │
               │                   (1)                                │
               └─view─file_name───────────────────────────────────────┘

►─Enter──────────────────────────────────────────────────────────────────►◄
```

Notes:

1 Valid with BIND 4.9.3 version of nslookup only.

Parameters

-Option
For a description of the **onslookup** options, see "onslookup options" on page 688.

-server_name
Directs the default name server to map *-server_name* to an IP address and then use 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 677. For v9 **nslookup** only, this can be a name that resolves to an IPv6 address. If the server exists on an IPv6-only host, the server name or address must be specified, as IPv6 addresses cannot be used in the resolver configuration file.

-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 677. For v9 **nslookup** only, this can be an IPv4 or an IPv6 address.

exit
Exits from **onslookup** interactive mode.

help or ?
Displays a brief summary of commands. Valid with version 4 of **nslookup** only.

host
host is the host name or Internet 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 Internet 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.

ls Lists various information available for the domain. By default, the IP address of each node in the domain is listed. This command is valid with version 4 of **nslookup** only. The parameters that can be used include:

ls_opt
To select resource records other than the default, specify one of the following options:

-t *querytype*
Lists all records of the specified type. See the *z/OS Communications Server: IP Configuration Reference* for detailed information about valid query types.

-a Lists aliases of hosts in the domain. (A synonym for `-t CNAME`.)

-d Lists all records for the domain. (A synonym for `-t ANY`.)

-h Lists CPU and operating system information for the domain. (A synonym for `-t HINFO`.)

-s Lists well-known services of hosts in the domain. (A synonym for `-t WKS`.)

domain
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 creates 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.

>|>> *file_name*
Output can be placed in a file for later viewing by specifying *file_name* for a specified domain.

> *file_name*
Creates the *file_name* to place output in. If *file_name* exists, the contents are overwritten.

>> *file_name*
Appends the output to the contents of *file_name*.

Note: There must be at least one space before and after the > or >> symbol.

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. For v9 **nslookup** only, 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.

root
Changes the default server to the server for the root of the domain name

space. The root server is `a.root-servers.net` by default, but can be changed using the `set root=name` subcommand. This command if valid only with version 4 of **nslookup**.

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
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. For v9 **nslookup** only, 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. In v9 mode, 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`. For v9, this is equivalent to the [no]search option. For v9, specifying the ′-domain=′ option will also cause the default domain name to be appended to the name being queried.

domain=*name*
This keyword is valid with version 4 of **nslookup** only. 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. For v4 only, 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 keyword `domain` can be abbreviated as `do`. For v9, this will also cause the -search option to be turned on. Also for v9, this will become the default search list and will override the default domain specified in the resolver configuration file.

[no]ignoretc
This keyword is valid with version 4 of **nslookup** only. Ignore packet truncation errors. The default is `noignoretc`. The keyword `ignoretc` can be abbreviated as `ig`.

port=*value*
Change the default TCP/UDP name server port to *value*. The default is 53. The keyword `port` can be abbreviated as `po`.

querytype=*type*

- For either version of **nslookup**, the keyword `querytype` can be abbreviated as `type`.
- For version 4 of **nslookup** only, the keyword `querytype` can also be abbreviated as `q`.
- For version 9 of **nslookup** only, 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`.

root=*name*
This keyword is valid with version 4 of **nslookup** only. Change the name of the root server to *name*. This keyword affects the `root` command. The default is `a.root-servers.net`. The keyword `root` can be abbreviated as `ro`.

[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`. For v9, this is equivalent to the [no]defname option.

srchlist=*name1***/***name2***/...**
This keyword is valid with version 4 of **nslookup** only. Change the default domain name to *name1* and the domain search list to *name1*, *name2*, and so on. A maximum of six names separated by slashes (/) can be specified. For example, `set srchlist=`*`lcs.MIT.EDU/ai.MIT.EDU/MIT.EDU`* sets the domain to `lcs.MIT.EDU` and the search list to the three names. This keyword overrides the default domain name and search list of the `set domain` command. Use the `set all` command to display the list. The default is based on hostname `/etc/resolv.conf` or LOCALDOMAIN. The keyword `srchlist` can be abbreviated as `srchl`.

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`.

view *file_name*
This command is valid with version 4 of **nslookup** only. It sorts and lists the output of *file_name*.

onslookup options

Purpose

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 677.

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.

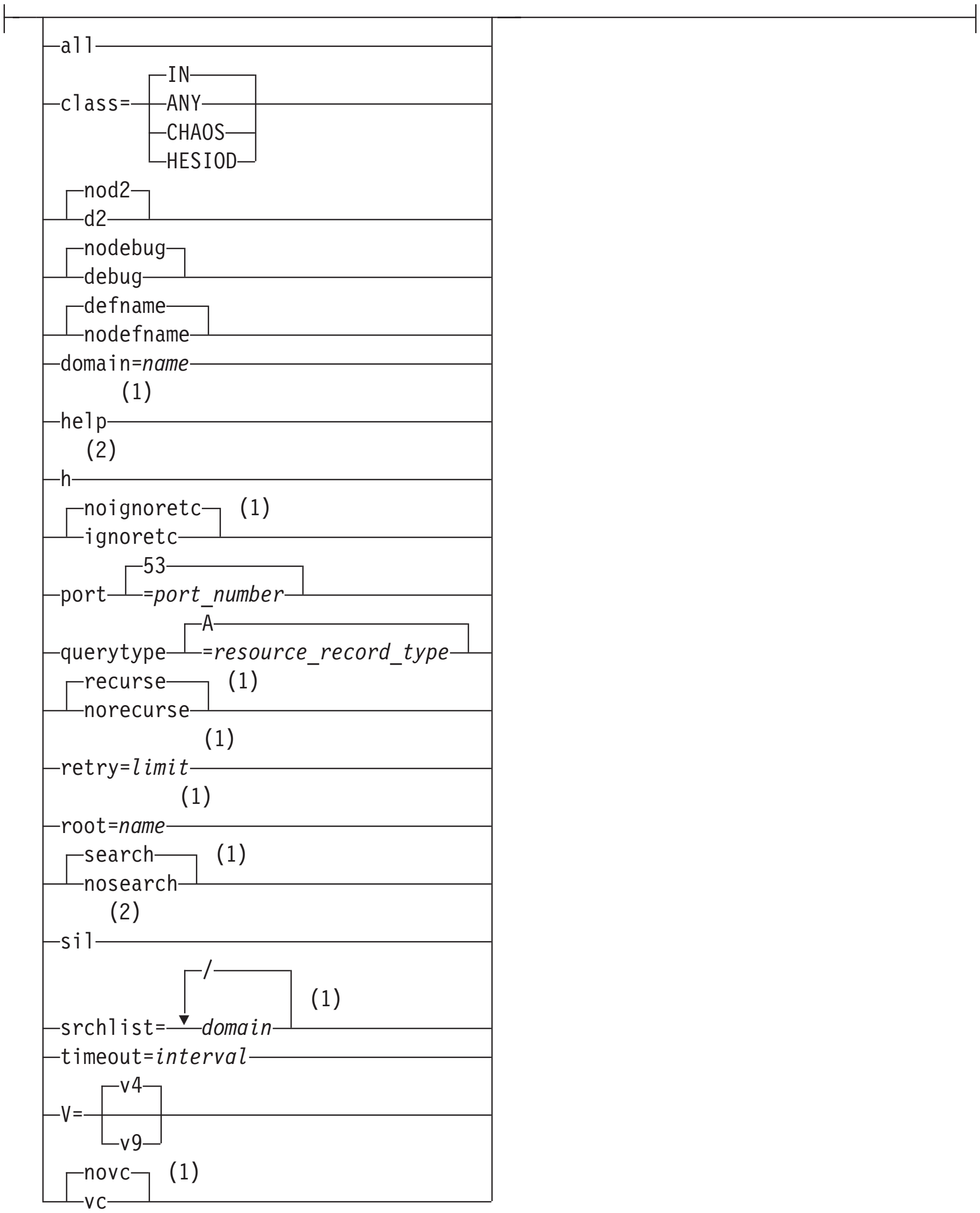
For example, to specify a name server (NS) type record lookup for the domain name `fourex.oz` in command mode you enter:

```
onslookup -querytype=ns fourex.oz
```

To submit the same request using interactive mode, enter the following sequence:

```
onslookup
set querytype=ns
fourex.oz
```

Format



Notes:

1 Valid with version v4 nslookup only.

2 Valid with version v9 nslookup only.

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. In v9 mode, 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. Valid with version 9 of **nslookup** only.

-help
Prints a brief summary of commands. Valid with version 4 of **nslookup** only.

-[no]ignoretc
This option is valid with version 4 of **nslookup** only. Ignore packet truncation errors. The default is `noignoretc`. The option `ignoretc` can be abbreviated as `ig`.

-port=*port_number*
Change the default TCP/UDP name server port to *port_number*. The default is 53. The option `port` can be abbreviated as `po`.

-querytype=*resource_record_type*

- For either version of **nslookup**, the option `querytype` can be abbreviated as `type`.
- For version 4 of **nslookup** only, the option `querytype` can also be abbreviated as `q`.
- For version 9 of **nslookup** only, 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`.

-root=*name*
This option is valid with version 4 of **nslookup** only. Change the name of the root server to *name*. This option affects the **root** command. The default is a.root-servers.net. The option root can be abbreviated as ro.

-[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. Valid with version 9 of **nslookup** only.

-srchlist=*name1***/***name2***/...**
This option is valid with version 4 of **nslookup** only. Change the default domain name to *name1* and the domain search list to *name1*, *name2*, and so on. A maximum of 6 names separated by slashes (/) can be specified. For example, set srchlist=*lcs.MIT.EDU*/*ai.MIT.EDU*/*MIT.EDU* sets the domain to lcs.MIT.EDU and the search list to the three names. This option overrides the default domain name and search list of the set domain command. Use the set all command to display the list. The default is based on hostname /etc/resolv.conf or LOCALDOMAIN. The option srchlist can be abbreviated as srchl.

-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 either BIND 9 mode or BIND 4.9.3 mode. The default is BIND 4.9.3 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.

Usage

The hyphen (-) operand is not valid preceding *options* in the .onslookuprc file. To make *querytype of NS* a default option for your **onslookup** commands, place one of the following statements in the .onslookuprc file:

- set querytype=ns
- querytype=ns

The optional .onslookuprc file (valid only with v4 **nslookup**) contains only **onslookup** options and defines the **onslookup** defaults. If the .onslookuprc file exists, the **onslookup** options are read from the file and executed before any queries are made. You must enter each option on a separate line. Blank lines are ignored.

The following is an example of the contents of the .onslookuprc file:

```
set domain=powers.oz
querytype=HINFO
set norecurse
vc
```

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. For v4 **nslookup**, the commands `set d2` and `set debug` produce identical output and are synonyms. For v9, `set d2` 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, only turns off *d2* and *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 proper 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) or RFC 2065 (for DNS 4.9.3) 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:

- Zones that are under dynamic control by **nsupdate** or a DHCP server should not be edited by hand. Manual edits could conflict with dynamic updates and cause data to be lost.
- The **nsupdate** command should not be used to update DNS zones that are managed by the automated domain name registration (ADNR) application. See Updates to an ADNR-managed zone in the *z/OS Communications Server: IP Configuration Guide* for more information.

nsupdate can work in BIND v4 or BIND v9 modes. **nsupdate** v4 commands can only send updates to a named v4 name server. **nsupdate** v9 commands can only send updates to a named v9 name server. **nsupdate** v9 commands should be used for IPv6 connections and update of IPv6 resource records.

- **BIND v4**

  You can use **nsupdate** 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 and specify a subcommand sequence in the -s command line parameter.

  The search order locations and order of priority from which the values for version BIND 4.9.3 **nsupdate** options can be specified are:

  1. **nsupdate** command options
  2. .nsupdaterc file in the home directory
  3. environment variable (LOCALDOMAIN)
  4. TCPIP.DATA statement values (See the *z/OS Communications Server: IP Configuration Guide* for the TCPIP.DATA search order for z/OS UNIX applications.)

  Tip: The **nsupdate** command for BIND 4.9.3 assumes that DDNS has already been configured for BIND 4. To enter interactive mode you must have a ddns.dat file that includes at least one entry with a primary domain name and host name (-p -h options) or you must specify the -p and -h options on the command line. For a successful dynamic update you must have a corresponding key pair (-g option) in the /etc/ddns.dat file.

- **BIND v9**

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 -V v9 /tmp/update.zone
```

The -V v9 option should be used on the command line (or the DNS_VERSION environment variable should be set to v9) and the file name should 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

Purpose

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. For v4 **nsupdate**, you can specify a subcommand sequence in the -s command line parameter. For v9 **nsupdate**, 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

BIND 4.9.3:

```
               ┌─────────────────────────────┐
               ▼                             │
├──nsupdate─────┬─────────────────────────┬────────────────────────────────────────────┤
               ├─ -k ──keyfile────────────┤
               ├─ -h ──hostname───────────┤
               ├─ -d ──domainname─────────┤
               ├─ -p ──primaryname────────┤
               ├─ -r ──IPaddress──────────┤
               ├─ -s ──"command string"───┤
               ├─ -a──────────────────────┤
               ├─ -g──────────────────────┤
               ├─ -q──────────────────────┤
               ├─ -v──────────────────────┤
               ├─ -?──────────────────────┤
               │          ┌─ v4─┐         │
               └─ -V──────┤     ├─────────┘
                          └─ v9─┘
```

BIND 9:

```
               ┌──────────────────────────────┐
               ▼                              │
├──nsupdate─────┬──────────────────────────┬───┬─────────────────┬──────────────────┤
               ├─ -d───────────────────────┤   └─batch_file_name─┘
               ├─ -v───────────────────────┤
               ├──┬─ -y ──keyname:secret─┬──┤
               │  └─ -k ──keyfile────────┘  │
               ├─ -D───────────────────────┤
               ├─ -M───────────────────────┤
               │          ┌─ v4─┐          │
               └─ -V──────┤     ├──────────┘
                          └─ v9─┘
```

Parameters

Valid with BIND 4.9.3

-k *keyfile*
By default, **nsupdate** uses /etc/ddns.dat for storing and retrieving key information. Use the -k parameter to specify an alternate key file.

-h *hostname*
Specifies the name or alias of a remote host for which the update will be made. When creating a KEY, it can also be the domain name for the zone SOA resource record.

-d *domainname*
Specifies the name of the dynamic domain for the host. Not needed if the -h parameter specifies a fully qualified host name.

-p *primaryname*
Specifies the fully qualified host name of the primary DDNS server.

-r *IPaddress*
Specifies the IP address used to update PTR records. (You cannot specify -r if you specify -h or -d.)

-s ″**command string**″
Command string is the string of subcommands with associated required input to be executed by **nsupdate**. Each subcommand is separated by a semicolon.

The following illustrates how the -s parameter is used to add an A Record with an expiration time of 1 hour (3600 seconds), and a signature 36 days (3110400 seconds):

```
nsupdate -h warpspeed.dynozone.sandbox -s a;9.67.96.10;s;36
>
The above example assumes that the key file entry for this host already exists
in /etc/ddns.dat and contains the fully qualified name of the
primary DDNS server.
```

-a The administrator mode flag. Specifies that the records in the update are to be signed and authenticated.

-g Only generates a key file entry for the host, but does not register the key with the DDNS server.

-q Specifies quiet mode. When this option is used, no prompting or informational messages are displayed.

-v Used for debugging purposes. When used on the command line, this turns on verbose mode. When verbose mode is on, all the requests to, and responses from, the name server are displayed.

-V Specifies the version of **nsupdate** (v4 or v9), and indirectly specifies the version of DDNS, since only v4 **nsupdate** can be used with v4 name servers, and v9 **nsupdate** can only be used with v9 name servers.

-? Displays help for **nsupdate**.

Tip: nsupdate for BIND 4.9.3 assumes that DDNS has already been configured for BIND 4. To enter interactive mode you must have a ddns.dat file that includes at least one entry with a primary domain name and host name (-p -h options) or you must specify the -p and -h options on the command line. For a successful dynamic update you must have a corresponding key pair (-g option) in the /etc/ddns.dat file.

Valid with BIND 9

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** (v4 or v9), and indirectly specifies the version of DDNS, since only v4 **nsupdate** can be used with v4 name servers, and v9 **nsupdate** can only be used with v9 name servers.

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 only be known 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

Purpose

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. There are two versions of subcommands available; one set is valid only with version BIND 4.9.3 (v4) of **nsupdate**, and one set for version BIND 9 (v9) of **nsupdate**.

Format

Start nsupdate subcommand mode

```
►►──nsupdate──Enter──────────────────────────────────────────────────►◄
```

Subsequent subcommand entry (valid with version 4 of nsupdate)

```
     ┌───────────────────┐
     ▼                   │
►►────────────────Enter──┴────────────────────────────────────────────►◄
      ├─quit────┤
      ├─add─────┤
      ├─delete──┤
      ├─exists──┤
      ├─new─────┤
      ├─sign────┤
      └─ttl─────┘
```

Subsequent subcommand entry (valid with version 9 of nsupdate)

```
     ┌─────────────────────────────────┐
     ▼                                 │
►►──────────────────────────────Enter──┴──────────────────────────────►◄
      ├─quit──────────────────────┤
      ├─prereq──┬─nxdomain──┬─────┤
      │         ├─yxdomain──┤     │
      │         ├─nxrrset───┤     │
      │         └─yxrrset───┘     │
      ├─server────────────────────┤
      ├─send──────────────────────┤
      ├─show──────────────────────┤
      ├─update──┬─add─────┬───────┤
      │         └─delete──┘       │
      └─zone──────────────────────┘
```

Parameters

The following describe valid v4 and v9 **nsupdate** command parameters.

- **Valid for v4 nsupdate**

  The following subcommands can be used in the **nsupdate** command shell and are valid only with version 4 of **nsupdate**. Some commands are for administrative purposes, others are update instructions. 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.

add Appends an ADD to the transaction list.

delete Appends a DELETE to the transaction list.

exists Appends an ADDNAMEEXIST to the transaction list.

new Appends an ADDNAMENEW to the transaction list.

sign Signs and sends the transactions to the transaction list.

ttl Sets the default TTL to be used in records on the update request. The default value is 4660.

- **Valid for v9 nsupdate**

  The following subcommands can be used in the **nsupdate** command shell and are valid only with version 9 of **nsupdate**. Some 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 will be 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***]**
  Specify the server name or IP address where all dynamic update requests will be 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 will send 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 Send update to server.

show Show 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 will be 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 v4 examples

The following are example console sessions using **nsupdate** BIND v4 in interactive mode. All examples assume the administrator has already set up a zone key and that the private key component for the zone is included in the local /etc/ddns.dat.

How an administrator removes and locks out a host name

The following example demonstrates an administrator's input and the system responses when removing and locking out a host name. Administrator input is highlighted in bold:

1. Generate a New Key for the Host

```
>nsupdate -g -h warpspeed.dynozone.sandbox -p
netadmin.dynozone.sandbox
--- NSUPDATE Utility ---
---
Key Gen ...... succeeded ...
```

2. Delete a User's A and KEY RRs and add a New KEY RR for New, Administrator Generated Key

```
nsupdate -a -h warpspeed.dynozone.sandbox -p
netadmin.dynozone.sandbox
--- NSUPDATE Utility ---

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> d
---
InitDDNSUpdate ...... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): a
....ip addr: *
DDNSUpdate_A (Delete *) ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> d
---
InitDDNSUpdate ...... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): key
DDNSUpdate_KEY DELETE *
succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> a
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): key
DDNSUpdate_KEY (Add Flags 0000 Protocol 0 Algid 1
      Keylen 64 Key0-150;: AQPS80e7uGuuNIdA ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> s
..sig Expiration (secs from now, ENTER for 3600):
..sig KEY pad (ENTER for default of 3110400):
DDNSSignUpdate ...succeeded
DDNSFinalizeUpdate ...succeeded
DDNSSendUpdate ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> q
```

How an administrator creates an alias for the dynamic zone

The following example demonstrates an administrator's input and the system responses when creating an alias for the dynamic zone. Administrator input is highlighted in bold:

```
>nsupdate -a -h ns-updates.dynozone.sandbox -p netadmin.dynozone.sandbox
--- NSUPDATE Utility ---

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
```

```
> a
---
InitDDNSUpdate ...... succeeded ...
..rrtype (A,PTR,CNAME,MX,KEY,HINFO): cname
....hostname: netadmin
DDNSUpdate_CNAME (Add netadmin.dynozone.sandbox) ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> s
..sig Expiration (secs from now, ENTER for 3600):
..sig KEY pad (ENTER for default of 3110400):
DDNSSignUpdate ...succeeded
DDNSFinalizeUpdate ...succeeded
DDNSSendUpdate ...succeeded

Enter Action (Add,Delete,Exists,New,TTL,Send,Quit)
> q
```

nsupdate BIND v9 examples

The following are 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. Since the `zone` and `server` subcommands were not explicitly issued, the defaults will come from the resolver configuration data set. Assume the following are 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 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
   ```

Return codes for version 4 of nsupdate

Return codes are used to give the user feedback on the progress of the command. Following are the return codes, origination of the return codes, and explanations (valid for version 4 of **nsupdate** only) for the most common problems that you might encounter:

Return code	Origin	Explanation
0	N/A	Successful.
-2	Local error	Input error.
-10	Local error	No key found in /etc/ddns.dat. A key is needed because either -f was specified or there is a KEY RR already in the name server data.
-11	Local error	Key in /etc/ddns.dat is not valid. Does not authenticate the user.
-12	Local error	No response received from the name server.
-1	Local error	Represents any other (local) error not specified above.
1	Server error	Format error. The name server was unable to interpret the request.
2	Server error	Server failure. The name server was unable to process this request because of a problem with the name server.
3	Server error	Name error. The domain name specified does not exist.
4	Server error	Not implemented. The name server does not support the specified operation code.
5	Server error	Refused. The name server refuses to perform the specified operation for security or policy reasons.
6	Server error	Alias error. A domain name specified in an update is an alias.

Return code	Origin	Explanation
7	Server error	Name Exists error. A name already exists. This return code is only meaningful from a server in response to an ADDNAMENEW operation.
8	Server error	Record error. Indicates that a resource record (RR) does not exist. This return code is only meaningful from a server in response to a DELETE operation.
9	Server error	Zone error. Indicates that the update is to be performed on a zone for which the server is not authoritative, or that the records to be updated exist in more than one zone.
10	Server error	Ordering error. If an ordering mechanism is used (for example, a SIG RR or an SOA RR), this code indicates an ordering error. Time-signed problems are also indicated by this return code.

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

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

Purpose

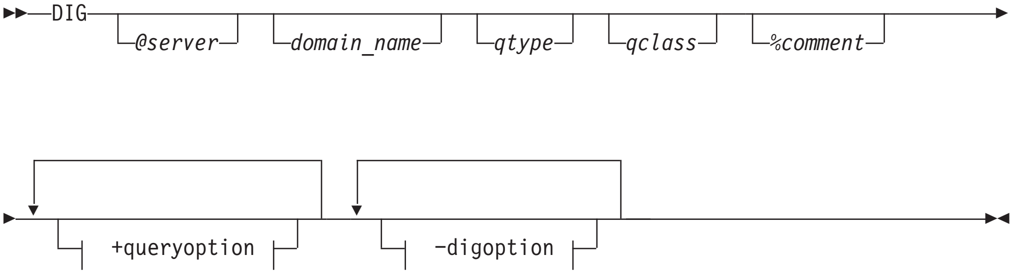
You can use DIG in command mode, where all options are specified on the invoking command line, or in batch mode, where a group of queries are placed in a data set and executed by a single invocation of DIG. DIG provides a large number of options for controlling queries and screen output, including most 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



+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 only use 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 will be 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 section).

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 should be 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 704. 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 704) 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 is an example of 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 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 704) 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 715, 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.
In the examples, all queries are issued from the MVS uluru system.

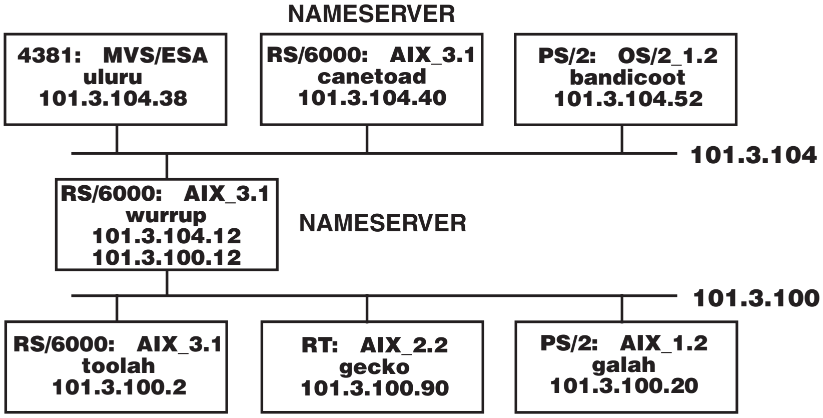


Figure 3. A TCP/IP network

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.

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

Purpose

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 should be 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 should be 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.

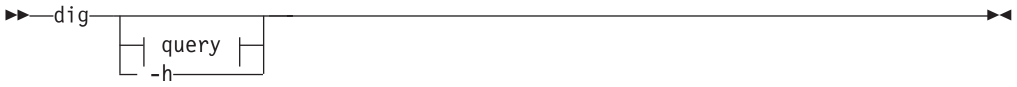
If a name server is not specified, **dig** will try each of the servers found in its TCPIP.DATA statements. When no command line arguments or options are given, **dig** will perform an NS query for ″.″ (the root).

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 on the search order for finding TCPIP.DATA statements. It uses directives from the resolver configuration file in the following order:

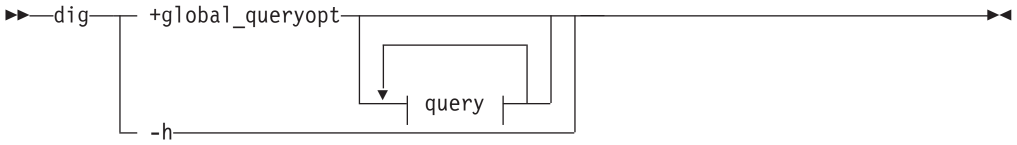
1. nameserver/nsinteraddr
2. options ndots:n
3. search
4. domain/domainorigin

Format

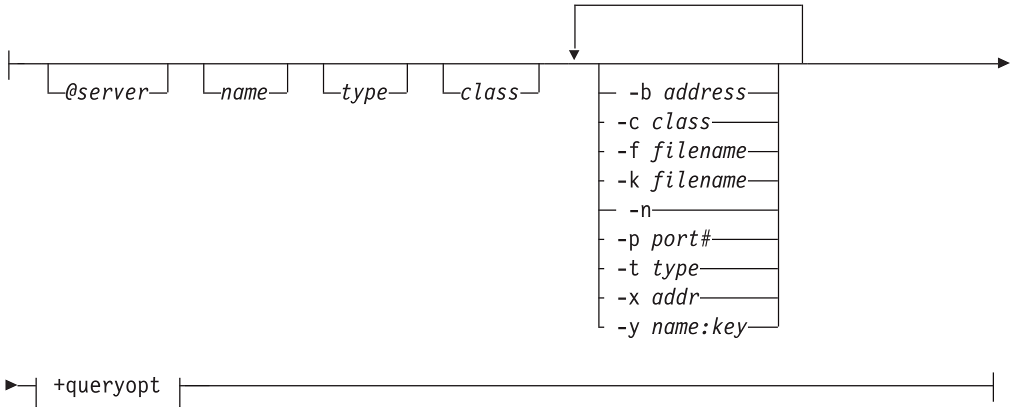
Command Line Mode



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 hostname, **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. In order to reach a name server on an IPv6-only host, this parameter must be provided, as the resolver configuration file does not support IPv6 name server addresses. 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. Each entry in the file should be organized 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 since 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. Caution should be taken 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 only recognized 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 retrying 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 desired) 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 will be 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 65535 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 will be 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 only apply 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 hostname
- 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

Purpose

Use the z/OS UNIX host command to identify the IP addresses associated with a specified DNS hostname or to identify the DNS hostnames associated with a specified IP address.

Format

```
►►──host──host──────────────────────────────────────────────────────────►◄
```

Parameters

host
DNS hostname 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

The z/OS UNIX hostname command is used to display the fully qualified DNS host name of the system. It supports several mechanisms for determining this information:

- TCPIP.DATA statement information
- DNS lookup on the value returned by gethostname()
- gethostname()

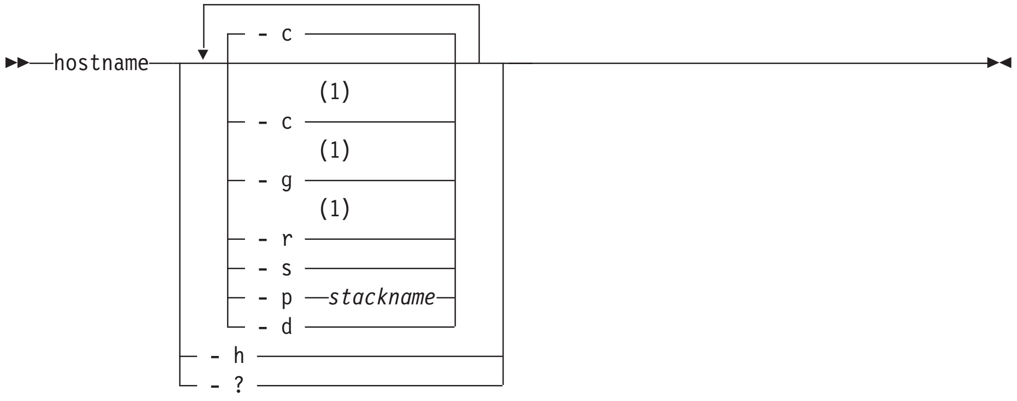
The z/OS UNIX hostname command must be issued from within the z/OS UNIX shell.

z/OS UNIX hostname—Identify the local host

Purpose

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).

-g uses gethostname() result.

-r uses DNS lookup on gethostname() result.

-s prints the short name of the host (without the DNS domain name).

-p *stackname*
uses this AF_INET stack.

-d prints trace messages for problem diagnosis.

-h displays the usage message.

-? displays the usage message.

Using the z/OS UNIX dnsdomainname command

The z/OS UNIX dnsdomainname command is used to display the DNS domain name of the system. It supports several mechanisms for determining this information:

- TCPIP.DATA statement information
- DNS lookup on the value returned by gethostname()
- gethostname()

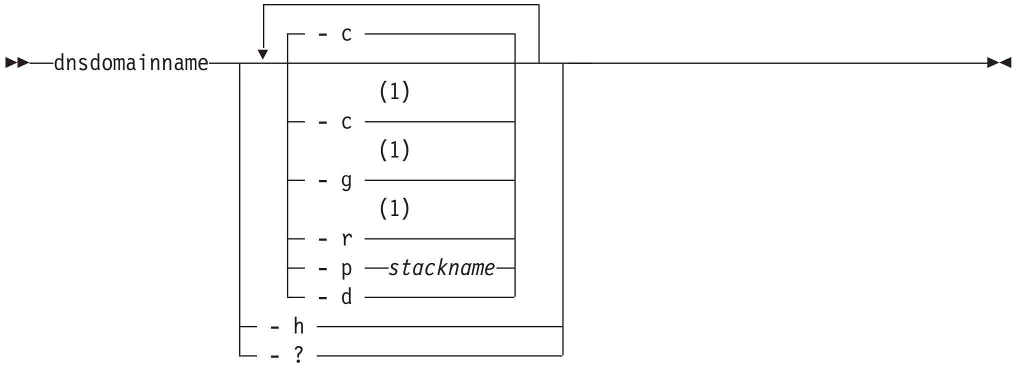
The z/OS UNIX dnsdomainname command must be issued from within the z/OS UNIX shell.

z/OS UNIX dnsdomainname—Display the DNS domain name

Purpose

Use the z/OS UNIX dnsdomainname 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 will be 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 737 for information on 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 should use dnsdomainname rather than domainname if this distinction is important.

Using the z/OS UNIX dnssec-keygen command

The **dnssec-keygen** command generates keys for DNSSEC, Secure DNS, as defined in RFC 2535. It also generates keys for use in Transaction Signatures (TSIG) which is defined in RFC 2845. The **dnssec-keygen** command can only be run from the z/OS UNIX shell.

If **dnssec-keygen** is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

dnssec-keygen—Generate key for DNSSEC

Purpose

DNSSEC and TSIG are used for BIND 9 DNS security features.

Format

```
►►──dnssec-keygen──┬─┤ Parameters ├─┬──────────────────────────────────────►◄
                   └─-h─────────────┘
```

Parameters:

```
├── -a algorithm── -e── -b keysize──┬─────────────────┬── -n nametype──────────►
                                    └─ -g generator───┘


   ┌────────────────────────┐
   ▼                        │
►──┬────────────────────────┬──name──────────────────────────────────────────┤
   ├─ -c class──────────────┤
   ├─ -p protocol-value─────┤
   ├─ -r randomdev──────────┤
   ├─ -s strength-value─────┤
   ├─ -t type───────────────┤
   └─ -v level──────────────┘
```

Parameters

-h Help, prints a short summary of the options and arguments to **dnssec-keygen**.

-a *algorithm*
The choice of encryption algorithm. This is a required entry.

algorithm must be one of the following:

DH Diffie-Hellman

DSA Digital Signature Algorithm

HMAC-MD5
HMAC-MD5

RSA Equivalent to RSAMD5

RSAMD5
Equivalent to RSA

Notes:

1. The argument identifying the encryption algorithm is not case sensitive.
2. DNSSEC RFC specifies DSA as a mandatory algorithm to implement and RSA as a recommended one.
3. Implementations of TSIG must support HMAC-MD5.

-b *keysize*
The number of bits in the key. This is a required entry.

The choice of key size depends on the algorithm used.

- RSA keys must be in the range 512–2048 bits.
- Diffie-Hellman keys must be in the range 128–4096 bits.

- DSA keys must be a multiple of 64 and in the range 512–1024 bits.
- HMAC-MD5 keys can be in the range 1–512 bits.

-c *class*
The class of the KEY record generated. The default is IN.

-e Tells **dnssec-keygen** to use a large exponent. Can only be used when generating RSA keys.

-g *generator*
Selects the Diffie-Hellman *generator* to be used. The only supported values value of *generator* are 2 and 5. If no Diffie-Hellman generator is supplied, a known prime from RFC 2539 will be used if possible; otherwise 2 will be used as the generator.

-n *nametype*
Specifies how the generated key will be used. This is a required entry.

nametype can be:

- ZONE
- HOST
- ENTITY
- USER

Notes:

1. In this context HOST and ENTITY are identical.
2. *nametype* is not case sensitive.

name
For zone signing, the domain name for which the key has to be generated. For TSIG security, an arbitrary name of your choice.

-p *protocol_value*
Sets the protocol value for the generated key to *protocol-value*. The default is

- 2 (E-mail) for keys of type USER
- 3 (DNSSEC) for all other key types

Note: Other possible values for this argument are listed in RFC 2535 and its successors.

-r *randomdev*
Specifies that file *randomdev* be the source of random data. If not specified, the user is prompted for keyboard input. The time interval between keystrokes and the Enter key is used to provide randomness.

Guideline: Do not use the randomdev file unless you can provide a different file of enough real random data for each new key generation. Instead, enter the required data using the keyboard. Furthermore, an otelnetd client is preferred over a TN3270 client. An otelnetd client sends a keystroke to the server each time a key is pressed, whereas the TN3270 client sends data to the server only when the Enter key is pressed. Because data entered using the otelnetd client generates more keystrokes (which are collected by the server) the final result should be more random.

-s *strength-value*
Sets the strength value for the key. The generated key will sign DNS resource records with a strength value of *strength value*. It should be a number in the range 0–15. The default strength is 0. The key strength field currently has no defined purpose in DNSSEC.

-t *type*

Indicates if the key is to be used for authentication or confidentiality. *type* can be one of:

AUTHCONF
The key can be used for authentication and confidentiality.

NOAUTHCONF
The key cannot be used for authentication or confidentiality.

NOAUTH
The key can be used for confidentiality, but not for authentication.

NOCONF
The key can be used for authentication, but not for confidentiality.

The default is AUTHCONF.

-v *level*

Sets the verbose level. As the debugging or tracing level increases, **dnssec-keygen** generates increasingly detailed reports. The default level is 0.

Usage

When **dnssec-keygen** completes it prints a character string in the form `Knnnn.+aaa+iiiii` on the standard output. This is an identification string for the key it has generated. These strings can be supplied as arguments to **dnssec-makekeyset**.

K Identifies the character string as a key.

nnnn. The dot-terminated domain name given by *name*.

+aaa The DNSSEC algorithm identifier.
- `001` for RSA
- `002` for Diffie-Hellman
- `003` for DSA
- `157` for HMAC-MD5

+iiiii A five-digit number identifying the key.

dnssec-keygen creates two files. The file names are adapted from the key identification string above. They have names of the form: `Knnnn.+aaa+iiiii.key` and `Knnnn.+aaa+iiiii.private`. These contain the public and private parts of the key respectively. The files generated by **dnssec-keygen** obey this naming convention to make it easy for the signing tool **dnssec-signzone** to identify which files have to be read to find the necessary keys for generating or validating signatures. The `.key` file contains a KEY resource record that can be inserted into a zone file with an $INCLUDE statement. The private part of the key is in the .private file. It contains details of the encryption algorithm that was used and any relevant parameters such as prime number, exponent, modulus, subprime. For obvious security reasons, this file does not have general read permission. The private part of the key is used by **dnssec-signzone** to generate signatures and the public part is used to verify the signatures. Both .key and .private key files are generated for symmetric encryption algorithm such as HMAC-MD5, even though the public and private key are equivalent. Domain names must be less than 236 characters because the generated suffix (`.+aaa+iiiii.private`) results in file names that are the maximum of 255 characters.

Examples

To generate a 768-bit DSA key for the domain example.com, the following command would be issued:

```
# dnssec-keygen -a DSA -b 768 -n ZONE example.com
Kexample.com.+003+26160
```

dnssec-keygen has printed the key identification string Kexample.com.+003+26160, indicating a DSA key with identifier 26160. It will also have created the files Kexample.com.+003+26160.key and Kexample.com.+003+26160.private containing the public and private keys for the generated DSA key.

Using the z/OS UNIX dnssec-makekeyset command

The **dnssec-makekeyset** command, used to configure the BIND 9 DNS DNSSEC feature, creates a key set file. A key set contains all of the keys containing KEY and SIG records for some zone which can then be signed by the zone's parent, if the parent zone is DNSSEC-aware. The **dnssec-makekeyset** command can only be run from the z/OS UNIX shell.

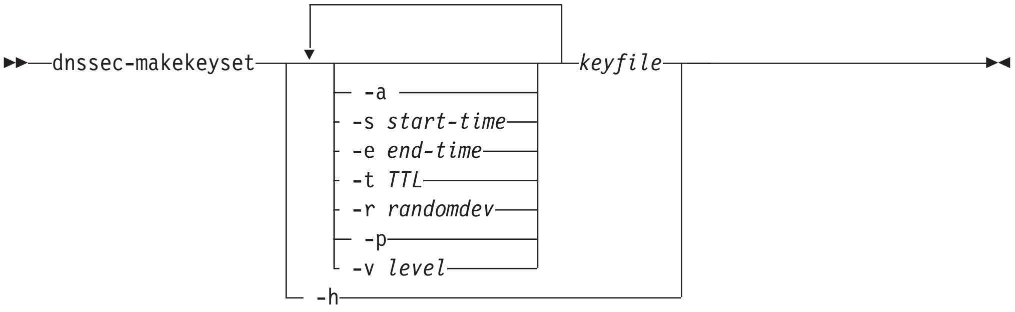
If **dnssec-makekeyset** is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

dnssec-makekeyset—Produce a set of DNSSEC keys

Purpose

The **dnssec-makekeyset** command is used to create a key set file from one or more keys created by the **dnssec-keygen** command.

Format



Parameters

keyfile

A key identification string as reported by **dnssec-keygen**. Multiple keyfile arguments can be supplied when there are several keys to be combined by **dnssec-makekeyset** into a key set.

The *keyfile* should be in the form `Knnnn.+aaa+iiiii` where `nnnn` is the name of the key, `aaa` is the encryption algorithm and `iiiii` is the key identifier. See “dnssec-keygen—Generate key for DNSSEC” on page 740 for details.

-a Verifies all generated signatures.

-s *start-time*

Sets the start time for any SIG records that are in the key set to become valid. *start-time* can either be an absolute or relative date.

- An absolute start time is indicated by a number in the format `YYYYMMDDHHMMSS`. For example: `20000530144500` denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *start-time* is given as `+N`, where `N` is the number of seconds from the current time.

If the -s option is not supplied, the current date and time is used for the start time of the SIG records.

-e *end-time*

Sets the expiration date for the SIG records. The expiration date specifies when the SIG records are no longer valid, not when they are deleted from caches on name servers. *end-time* can either be an absolute or relative date.

- An absolute end time is indicated by a number in the format `YYYYMMDDHHMMSS`. For example: `20000530144500` denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *end-time* is given as `+N`, where `N` is the number of seconds from the current time.
- If *end-time* is written as `now+N`, the SIG records will expire in `N` seconds after the current time.

When no *end-time* is set for the SIG records, **dnssec-makekeyset** defaults to an expiration time of 30 days from the start time of the SIG records.

-t *TTL*
Sets a time-to-live (*TTL*) value that is assigned to the assembled KEY and SIG records in the output file. *TTL* is expressed in seconds. If not provided, **dnssec-makekeyset** prints a warning and uses a default *TTL* of 3600 seconds.

-r *randomdev*
Specifies that file *randomdev* be the source of random numbers. If not specified, the user is prompted for keyboard input. The time interval between keystrokes and the Enter key is used to provide randomness.

Note: Using the randomdev file is not recommended unless you can provide a different file of enough real random data for each new use. Instead, use the keyboard for entropy. Furthermore, an otelnetd client is preferred over a TN3270 client since the entropy is gathered on the time between keystrokes. An otelnetd client sends the keystroke to the server each time it is pressed, whereas the TN3270 client only sends data to the server when the enter key is pressed.

-p Use pseudo-random data when self-signing the keyset. This is faster, but less secure, than using genuinely random data for signing. This option can be useful when the entropy source is limited.

-v *level*
Sets the verbose level. As the debugging or tracing level increases, **dnssec-makekeyset** generates increasingly detailed reports. The default level is 0.

-h Help, prints a short summary of the options and arguments to **dnssec-makeyset**.

Usage

When successful, **dnssec-makekeyset** creates `keyset-nnnn.` file name . This file contains the KEY and SIG records for domain `nnnn`. This is the domain name part from the key file identifier produced when **dnssec-keygen** created the domain's public and private keys. The **dnssec-makekeyset** command groups these keys together, adds TTLs and expirations dates to the keys, and signs the zone's public keys with the zones private keys. The `keyset-nnnn.` file can then be transferred to the DNS administrator of the parent zone for them to sign the contents with **dnssec-signkey**.

Examples

The following command generates a key set for the DSA key for `example.com`.

```
# dnssec-makekeyset -t 86400 -s 20000701120000 -e +2592000 Kexample.com.+003+26160
```

dnssec-makekeyset creates a file called `keyset-example.com.` containing a SIG and KEY record for `example.com`. These records will have a TTL of 86400 seconds (1 day). The SIG record becomes valid at noon UTC on July 1st 2000 and expires 30 days (2592000 seconds) later.

The DNS administrator for *example.com* could then send `keyset-example.com.` to the DNS administrator for *.com* to sign the resource records in the file. This assumes that the *.com* zone is DNSSEC-aware and the administrators of the two zones have some mechanism for authenticating each other and exchanging the keys and signatures securely.

Using the z/OS UNIX dnssec-signkey command

The **dnssec-signkey** command, used to configure DNSSEC security features for a BIND 9 name server, signs one child's keyset with the parent zone's private key. The **dnssec-signkey** command can only be run from the z/OS UNIX shell.

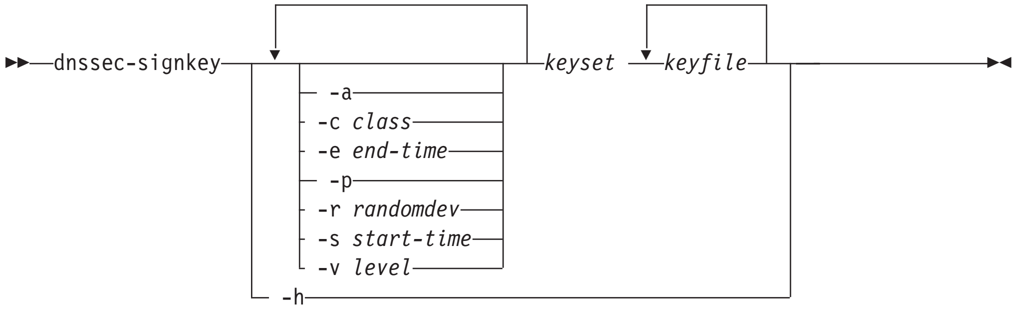
If **dnssec-signkey** is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments.

dnssec-signkey—DNSSEC keyset signing tool

Purpose

dnssec-signkey is used to sign a key set for a child zone. Typically this would be provided by a .keyset file generated by **dnssec-makekeyset**. This provides a mechanism for a DNSSEC-aware zone to sign the keys of any DNSSEC-aware child zones. The child zone's key set gets signed with the zone keys for its parent zone.

Format



Parameters

keyset
The pathname of the child zone's `keyset-` file.

keyfile
A key identification string as reported by **dnssec-keygen** for the parent zone. This allows the child's keys to be signed by more than one parent zone key.

-a Verifies all generated signatures.

-c *class*
Defines the class of the generated resource records. The default is IN.

-e *end-time*
Sets the expiration date for the SIG records. The expiration date specifies when the SIG records are no longer valid, not when they are deleted from caches on name servers. *end-time* can either be an absolute or relative date.

- An absolute end time is indicated by a number in the format `YYYYMMDDHHMMSS`. For example: `20000530144500` denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *end-time* is given as `+N`, where `N` is the number of seconds from the current time.
- If *end-time* is written as `now+N`, the SIG records will expire in `N` seconds after the current time.

When no *end-time* is set for the SIG records, **dnssec-signkey** defaults to an expiration time of 30 days from the start time of the SIG records.

-r *randomdev*
Specifies that file *randomdev* be the source of random numbers. If not specified, the user is prompted for keyboard input. The time interval between keystrokes and the Enter key is used to provide randomness.

Note: Using the randomdev file is not recommended unless you can provide a different file of enough real random data for each new use. Instead, use the keyboard for entropy. Furthermore, an otelnetd client is preferred over a TN3270 client since the entropy is gathered on the time between keystrokes. An otelnetd client sends the keystroke to the server each time it is pressed, whereas the TN3270 client only sends data to the server when the enter key is pressed.

-s *start-time*
Sets the start time for any SIG records that are in the key set to become valid. *start-time* can either be an absolute or relative date.

- An absolute start time is indicated by a number in the format `YYYYMMDDHHMMSS`. For example: `20000530144500` denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *start-time* is given as `+N`, where `N` is the number of seconds from the current time.

If the -s option is not supplied, the current date and time is used for the start time of the SIG records.

-p Use pseudo-random data when self-signing the keyset. This is faster, but less secure, than using genuinely random data for signing. This option can be useful when the entropy source is limited. It could also be used for short-lived keys and signatures that do not require as much protection against cryptanalysis, such as when the key will be discarded long before it could be compromised.

-v *level*
Sets the verbose level. As the debugging or tracing level increases, **dnssec-signkey** generates increasingly detailed reports. The default level is 0.

-h Help, prints a short summary of the options and arguments to **dnssec-signkey**.

Usage

When **dnssec-signkey** completes successfully, it generates a file called `signedkey-nnnn.` containing the signed keys for child zone `nnnn`. The keys from the keyset file will have been signed by the parent zone's key or keys which were supplied as keyfile arguments. This file should be sent to the DNS administrator of the child zone. The DNS administrator arranges for its contents to be incorporated into the zone file when it next gets signed with **dnssec-signzone**. A copy of the generated `signedkey` file should be kept by the parent zone's DNS administrator, since it will be needed when signing the parent zone.

Examples

The DNS administrator for a DNSSEC-aware `.com` zone would use the following command to make **dnssec-signkey** sign the keyset file for `example.com`:

```
# dnssec-signkey keyset-example.com. Kcom.+003+51944
```

where `Kcom.+003+51944` was a key file identifier produced when **dnssec-keygen** generated a key for the `.com` zone. **dnssec-signkey** produces a file called `signedkey-example.com.` which has the keys for `example.com` signed by the com zone's zone key.

Using the z/OS UNIX dnssec-signzone command

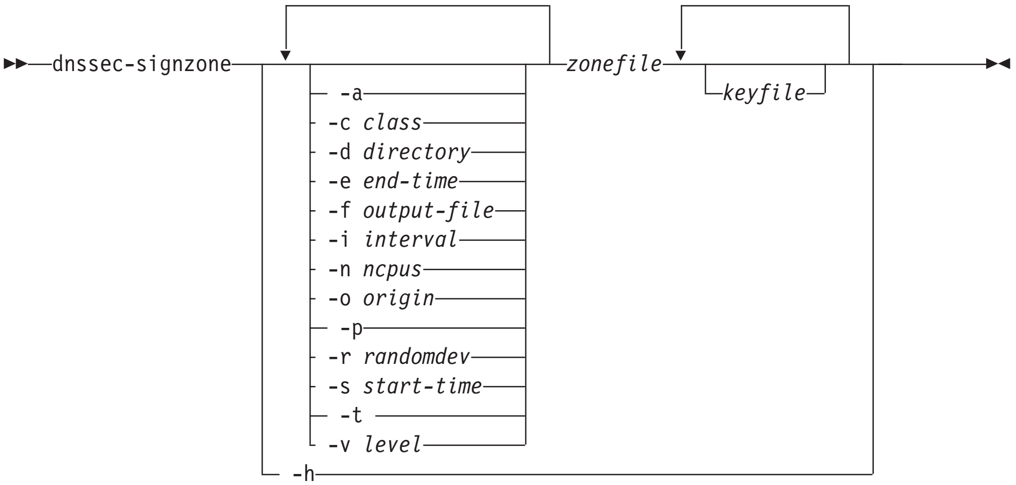
The **dnssec-signzone** command, used to configure the DNSSEC security feature for BIND 9 name servers, signs zones with the keys generated by dnssec-keygen. By signing zones with a private key, users of that data which have the public key or can securely obtain the public key can be assured that the data is authentic. The dnssec-signzone command can only be run from the z/OS UNIX shell.

dnssec-signzone—DNSSEC zone signing tool

Purpose

Used to sign a zone.

Format



Parameters

zonefile
`zonefile` is the name of the unsigned zone file. Use this option unless the file name is the same as the name of the zone. *origin* will be the fully qualified domain origin for the zone.

-a Verify the signatures generated. **dnssec-signzone** does not verify the signatures by default.

-c *class*
Defines the class of the generated resource records. The default is IN.

-d*directory*
The directory specified by this option will be searched for the signedkey files. The default is the current directory.

-i *interval*
Specifies the interval period as an offset from the current time (in seconds). When a previously signed zone is passed as input to dnssec-signzone, records might be resigned. Whether or not to resign records is configurable by using this option. If a SIG record expires after the interval period, it is retained. Otherwise, it is considered to be expiring soon, and **dnssec-signzone** will remove it and generate a new SIG record to replace it. The default interval period is one quarter of the difference between the specified signature end and start dates. So if the -e and -s options are not specified, **dnssec-signzone** generates signatures that are valid for 30 days from the current date by default, with a interval period of 7.5 days. Therefore, if any SIG records are due to expire in less than 7.5 days, they will be replaced with new ones.

-e *end-time*
Sets the expiration date for the SIG records. The expiration date specifies when the SIG records are no longer valid, not when they are deleted from caches on name servers. *end-time* can either be an absolute or relative date.

- An absolute end time is indicated by a number in the format YYYYMMDDHHMMSS. For example: 20000530144500 denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *end-time* is given as +N, where N is the number of seconds from the current time.
- If *end-time* is written as now+N, the SIG records will expire in N seconds after the current time.

When no *end-time* is set for the SIG records, **dnssec-signzone** defaults to an expiration time of 30 days from the start time of the SIG records.

-f *output-file*
Creates the *output-file* file that contains the signed zone file. The default is zonefile.signed.

-n *ncpus*
Determines the number of threads to use while running the program. The default is one thread.

-o *origin*
Zone origin (name of zonefile)

-p Use pseudo-random data when self-signing the keyset. This is faster, but less secure, than using genuinely random data for signing. This option can be useful when the entropy source is limited. It could also be used for short-lived keys and signatures that do not require as much protection against cryptanalysis, such as when the key will be discarded long before it could be compromised.

-r *randomdev*
Specifies that file *randomdev* be the source of random numbers. If not specified, the user is prompted for keyboard input. The time interval between keystrokes and the Enter key is used to provide randomness.

Note: Using the randomdev file is not recommended unless you can provide a different file of enough real random data for each new use. Instead, use the keyboard for entropy. Furthermore, an otelnetd client is preferred over a TN3270 client since the entropy is gathered on the time between keystrokes. An otelnetd client sends the keystroke to the server each time it is pressed, whereas the TN3270 client only sends data to the server when the enter key is pressed.

-s *start-time*
Sets the start time for any SIG records that are in the key set to become valid. *start-time* can either be an absolute or relative date.

- An absolute start time is indicated by a number in the format YYYYMMDDHHMMSS. For example: 20000530144500 denotes 14:45:00 UTC on May 30th, 2000.
- A relative start time is supplied when *start-time* is given as +N, where N is the number of seconds from the current time.

If the -s option is not supplied, the current date and time is used for the start time of the SIG records.

-t Prints statistics after signing the zone.

-v *level*
Sets the verbose *level*. The default is 0.

keyfile
Each `keyfile` argument would be an identification string for a key created with **dnssec-keygen**(8). If the zone to be signed has any secure subzones, the .signedkey files for those subzones need to be available in the current working directory used by **dnssec-signzone**, or in the directory specified by the -d option.

-h Help, prints a short summary of the options and arguments to **dnssec-signzone**.

Usage

Any signedkey files for the zone to be signed should be present in the current directory, along with the keys that will be used to sign the zone. If no keyfile arguments are supplied, the default behavior is to use all of the zone's keys that are present in the current directory. Providing specific keyfile arguments constrains **dnssec-signzone** to only use those keys for signing the zone.

dnssec-signzone will generate NXT and SIG records for the zone and produce a signed version of the zone. If there is a signedkey file from the zone's parent, the parent's signatures will be incorporated into the generated signed zone file. The security status of delegations from the signed zone (whether the child zones are DNSSEC-aware or not) is set according to the presence or absence of a signedkey file for the child.

Examples

The following example shows how **dnssec-signzone** could be used to sign the zone file. The zone file for this zone is `example.com`, which is the same as the origin, so there is no need to use the -o option to set the origin. This zone file contains the keyset for `example.com` that was created by **dnssec-makekeyset**. The zone's keys were either appended to the zone file or incorporated using a $INCLUDE statement. If there was a signedkey file from the parent zone (`signedkey-example.com.`), it should be present in the current directory. This allows the parent zone's signature to be included in the signed version of the `example.com` zone.

```
# dnssec-signzone example.com Kexample.com.+003+26160
```

dnssec-signzone will create a file called `example.com.signed`, the signed version of the `example.com` zone. This file can then be referenced in a zone{} statement in /etc/named.conf so that it can be loaded by the name server.

Using the z/OS UNIX dnsmigrate command

The **dnsmigrate** command is a migration aid that will convert `named.boot` files for the BIND 4.9.3 mode, into `named.conf` files suitable for the BIND 9 mode. The **dnsmigrate** command can only be run from the z/OS UNIX shell.

dnsmigrate—Configuration file migration

Purpose

The **dnsmigrate** utility is designed to aid in the migration from BIND 4.x.x style bootfiles to BIND 9.x.x's newer syntax. The utility parses the contents of the source bootfile and writes the corresponding directives into a valid 9.x.x configuration file. If the source file contains any directives which are obsolete the utility exits with an error message directing the user to remove all deprecated syntax from the source file. This utility only converts the named configuration files; changes that are necessary for the actual zone files (e.g., adding a $TTL directive) must be made manually.

Format

```
►►──dnsmigrate──┬─────────────────┬──────────────────────────────────────────────►◄
                ├─ -i input_file──┤
                └─ -o output_file─┘
```

Parameters

-i *input_file*
Specifies the bootfile to convert. The default is /etc/named.boot.

-o *output_file*
Specifies the destination file for the converted bootfile. If the specified file already exists the user is prompted to overwrite it. The default is /etc/named.conf.

Examples

The following command, using **dnsmigrate** without parameters, converts the default input file (`/etc/named.boot`) and writes the results to the default output file (`/etc/named.conf`).

```
dnsmigrate
```

The following converts the specified input file (`/tmp/named.boot`) to the specified output file (`/tmp/named.conf`). Note that it does not matter what order the input and output file options are used.

```
dnsmigrate -o /tmp/named.conf -i /tmp/named.boot
```

Using the z/OS UNIX rndc command

Remote Name Daemon Control (rndc) command allows the system administrator to control the operation of a name server. If rndc is invoked with no command line options or arguments, it prints a short summary of the supported commands and the available options and their arguments. rndc can only be used with the v9 (BIND 9) name server and will not function with the v4 (BIND 4.9.3) name server.

The function of rndc can be used as a secure remote client to control the name server. Some local UNIX signal functions for the name server can be replaced by equivalent rndc functions. The name server and rndc communicates over a TCP connection, sending commands authenticated with digital transaction signatures (TSIG). This provides TSIG-style authentication for the command request and the name server's response. All commands sent over the channel must be signed by a key_id known to the server. Therefore, rndc and the name server must be configured with a shared-secret. This shared-secret is a TSIG key, which can be generated with the dnssec-keygen utility. The only supported encryption algorithm for rndc is HMAC-MD5, which uses a shared-secret on each end of the connection.

An rndc.conf-style file named `rndc.key` can be generated using the rndc-confgen program with the -a parameter. The name server will use this file if the controls{} statement is not present in `named.conf`, or if the controls{} statement is present but there is no *keys* clause within it. The rndc program will first look for `/etc/rndc.conf`, and if not found, search for `/etc/rndc.key`.

Note: If rndc is to be used on a remote host, the `rndc.key` file will have to be copied to that host. Since the file contains a shared-secret key, the file should be moved securely and the file permissions set accordingly.

See the *z/OS Communications Server: IP Configuration Guide* for information about rndc configuration (rndc.conf file).

rndc—Remote control of name server

Purpose

Use the z/OS UNIX rndc command to control functions of the remote name server.

Note: Authorization to use the rndc tool depends on the target BIND 9 name server configuration file control statement. See the *z/OS Communications Server: IP Configuration Guide* for information about TCPIP.DATA statements.

Format

```
►►──rndc──┬──────────────┬──command──command ...──────────────────────►◄
          ├─ -c ──config─┤
          ├─ -s ──server─┤
          ├─ -p ──port───┤
          ├─ -y ──key────┤
          └─ -V──────────┘
```

Parameters

-c *config*
: Used to specify an alternate configuration file. rndc reads its default configuration file, /etc/rndc.conf to determine how to contact the name server and decide what algorithm and keys to use.

-s *server*
: Specifies the host name or IP address of the name server. An IPv4 or IPv6 address or a name that resolves to an IPv4 or IPv6 address can be specified.

server is the name or address of the server which matches a server{} statement in the configuration file for rndc. If no server is supplied on the command line, the host named by the default-server clause in the options{} statement of the configuration file will be used.

Restriction: You cannot specify scope information with the host name or the IP address of the name server.

-p port
: Specifies the rndc control port the name server is listening on.

The -p option can be used to make rndc send commands to TCP port number *port* on the system running the name server instead of BIND 9's default control channel port of 953.

-y *key*
: Identifies the key_id to use from the configuration file. The key_id must be known by the name server with the same algorithm and secret string in order for control message validation to succeed. If no -y option is provided, rndc will first look for a key clause in the server{} statement of the server being used, or if no server{} statement is present for that host, then the default-key clause of the options{} statement.

-V Verbose.

command *command . . .*
: command is one of the following for named:

dumpdb
: Dump any caches to the dump file (named_dump.db).

flush Flushes the server's cache.

halt Stop the server without updating master zone files with latest dynamic updates recorded in journal files.

notrace
Sets the server's debugging level to 0. This only affects the *dynamic* debug level. Thus, only the *default-debug* logging channel and any user-defined logging channel which uses the `severity dynamic` statements will be affected.

querylog
Toggle query logging.

reconfig
Reload the configuration file and load new zones, but do not reload existing zone files even if they have changed. This is faster than a full reload when there is a large number of zones because it avoids the need to examine the modification times of the zones files.

refresh zone [*class* [*view*]]
Schedule immediate maintenance for a zone.

reload Reload configuration file and zones.

reload zone [*class* [*view*]]
Reload a single zone.

Note: Because the dynamic zones should not be manually edited, the **rndc reload** and **rndc reload zone** commands will have no effect on them, regardless of whether you incremented the zone's SOA value.

stats Write server statistics to the statistics file.

status Display status of the server.

Note: The `number of zones` displayed is calculated according to the following table. The names `Master`, `Slave`, `Stub`, `Hint`, and `Forward` are zone types. *version.bind* and *authors.bind* are CHAOS class zones, which are added to the name server automatically. *version.bind* is always added and *authors.bind* might or might not be added.

Table 17. Zones counted for number of zones

Counted	**Not counted**
Master zones	Hint zones
Slave zones	Forward zones
Stub zones	
version.bind	
authors.bind (if no 'version' option is present)	authors.bind (if 'version' option is present)

stop Stop the server after updating master zone files with latest dynamic updates recorded in journal files.

trace Increment the servers debugging level by one. This only affects the *dynamic* debug level. Thus, only the *default-debug* logging channel and any user-defined logging channel which uses the `severity dynamic` statements will be affected.

trace *level*

Sets the server's debugging level to an explicit value. This only affects the *dynamic* debug level. Thus, only the *default-debug* logging channel and any user-defined logging channel which uses the `severity dynamic` statements will be affected.

rndc-confgen — rndc key generation tool

Purpose

The command **rndc-confgen** generates configuration files for rndc. It can be used as a convenient alternative to writing the *rndc.conf* file and the corresponding controls and key statements in *named.conf* by hand. Alternatively, it can be run with the -a option to set up a *rndc.key* file and avoid the need for a *rndc.conf* file and a controls statement.

Format

```
►►──rndc-confgen──┬─────────────────┬─────────────────────────────────────►◄
                  ├─ -a─────────────┤
                  ├─ -b keysize─────┤
                  ├─ -c keyfile─────┤
                  ├─ -h─────────────┤
                  ├─ -k keyname─────┤
                  ├─ -p port────────┤
                  ├─ -r randomdev───┤
                  ├─ -s address─────┤
                  ├─ -t chrootdir───┤
                  └─ -u user────────┘
```

Parameters

-a Provides automatic rndc configuration. This creates a file *rndc.key* in /etc that is read by rndc and on named startup. The *rndc.key* file defines a default command channel and authentication key allowing rndc to communicate with `named` with no further configuration.

Allows BIND 9 and rndc to be used as drop-in replacements for BIND 8 and ndc, with no changes to the existing BIND 8 named.conf file.

-b Specifies the size of the authentication key in bits. It must be in the range 1–512. The default is 128.

-c Used with the -a option to specify an alternate location for *rndc.key*.

-h Prints a short summary of the options and arguments for **rndc-confgen**.

-k Specifies the key name of the rndc authentication key. This must be a valid domain name. The default is rndc-key.

-p Specifies an alternate command channel port, where named listens for connections from rndc. The default port is 953.

-r Specifies a source of random data for generating the authorization. The default source for z/OS UNIX is entered from the keyboard. The argument for this option specifies the name of a character device or file containing random data to be used instead of the default. The special value keyboard indicates that keyboard input should be used.

-s Specifies the IP address where named listens for command channel connections from rndc. The default is the loopback address 127.0.0.1. This can be an IPv4 or an IPv6 address.

-t Used in conjunction with the -a option to specify a directory where `named` will run `chrooted`. An additional copy of the *rndc.key* will be written relative to this directory so that it will be found by the `chrooted` `named`.

-u Used in conjunction with the -a option to set the owner of the file generated. If -t is also specified, only the file in the chroot area has its owner changed.

Examples

To allow rndc to be used with no manual configuration, run:

```
rndc-confgen -a
```

To print a sample rndc.conf file and corresponding controls and key statements to be manually inserted into named.conf, run

```
rndc-confgen
```

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

Purpose

The z/OS UNIX **snmp** command provides the SNMP manager function from the z/OS shell to query SNMP agents for network management information.

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.

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

```
            ┌─ -d 0──────────┐  ┌─ -h localhost──┐  ┌─ -r 2───────────┐
►►──snmp────┼────────────────┼──┼────────────────┼──┼─────────────────┼──────────────►
            └─ -d debug_level┘  └─ -h target host┘  └─ -r retry number┘

    ┌─ -c public────────┐  ┌─ -t 3───────────┐
►───┼───────────────────┼──┼─────────────────┼──┬────┬──┬────┬──set──────────►
    └─ -c community_name┘  └─ -t timeout value┘  └─ -v┘  └─ -a┘
```

```
   ┌──────────────────────────────────┐
   ▼                                  │
►──mib_variable──┬─────────┬──value───┴──────────────────────────────────────►◄
                 └─vartype─┘
```

Walking the MIB Tree

```
                ┌─ -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──┘

►──┬─walk─────────────────────────────────────────────────────────┬─mib_variable───►◄
   │  ┌─ -m 10─────────────────┐   ┌─ -n 0─────────────┐           │
   └──┴─ -m max repetitions────┴───┴─ -n non-repeaters─┴─bulkwalk──┘
```

Displaying snmp Help

```
►►──snmp── -?──────────────────────────────────────────────────────────────────►◄
```

Receiving a Trap

```
                ┌─ -d 0───────────┐   ┌─ -p 162────────┐
►►──snmp────────┴─ -d debug_level─┴───┴─ -p port_number─┴─trap────────────────────►◄
```

Finding a MIB Variable Name

```
                ┌─ -d 0───────────┐
►►──snmp────────┴─ -d debug_level─┴──findname──mib_variable───────────────────────►◄
```

Parameters

-d *debug_level*

Specifies the debug level. The default level is 0, which means no debug. The higher the debug level, the greater the amount of messages that are displayed. The debug levels are 0–4.

-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 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. However, when using a community defined by the SNMP_COMMUNITY statement, both the *-c community_name* and the *-h target host* values must be specified.

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 should be 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 should be 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 only supports 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 names 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 file that snmp uses to determine whether it should send an SNMPv1, SNMPv2c, or SNMPv3 request is the OSNMP.CONF file. 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" on page 774.

For more sophisticated management support, consider using the AON support provided in *Tivoli NetView for z/OS Version 5.1*, which provides panels-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

Purpose

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 retries the request 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

Return code	Description
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:

  ```
  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:

```
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:

```
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:

```
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:

```
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 839 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 desired 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:

```
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 should receive that agent's traps. The host name of your system should 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:

  ```
  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:

  ```
  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:

```
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 retries 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 777, 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 desired, you should 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 798.

For a list of variables supported by the z/OS Communications Server IP agent, see Appendix B, "Management Information Base (MIB) objects," on page 839.

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

The following are 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	Internet 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

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 will be only 16 bytes long.

Using the pwtokey facility

Purpose

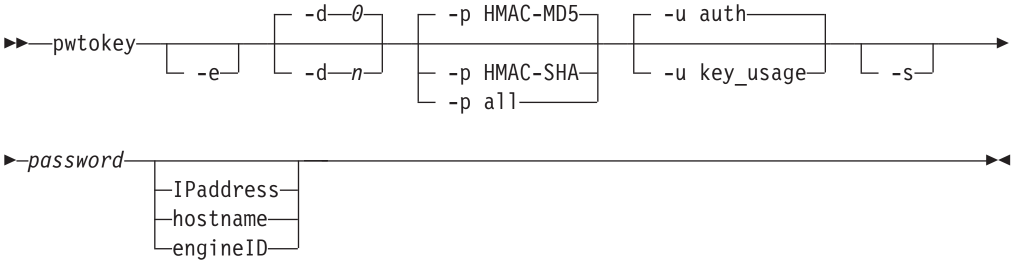
z/OS Communications Server provides a facility called pwtokey that allows conversion of passwords into localized and nonlocalized authentication and privacy keys, for SNMP or OMPROUTE.

- 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, CBC 56-bit DES encryption requires the use of 32 hexadecimal digit (16 byte) keys. However, if the key is generated using HMAC-SHA, which produces keys 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



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 only applicable when generating keys for SNMP.

-d *n*
This flag indicates what level of debug information is desired. 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 should be generated. Valid values are:

HMAC-MD5
Generates keys for use with the HMAC-MD5 authentication protocol. This is the only protocol that should be used 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 should be 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.

Notes:

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). Since 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 for DES.

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

Purpose

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

```
                         ┌─-d 0─┐  ┌─-p HMAC-MD5─┐  ┌─-u auth──────┐
►►──pwchange──┬──────┬───┼──────┼──┼─────────────┼──┼──────────────┼──┬────┬──────────────►
              └─-e───┘   └─-d n─┘  └─-p protocol─┘  └─-u key_usage─┘  └─-s─┘

►─old_password──new_password──┬─IPaddress─┬────────────────────────────────────────────►◄
                              ├─hostname──┤
                              └─engineID──┘
```

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 desired. 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 should be 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 should be 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

Purpose

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.

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 839. 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. 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.

This section lists the main areas of the TCP/IP stack for which MIB data has been defined, and a description of the Subagent support for the MIB data in each area.

For some of these standard MIBs, the MIB data is defined in IETF Internet drafts instead of RFCs. These Internet drafts provide version-neutral MIB tables that can support both IPv4 and IPv6 data. The IPv4-only MIB data in these drafts has been deprecated. This means that this MIB data is still supported, but a management application should not implement new support for this MIB data. Instead, management applications should implement support for the version-neutral MIB data.

The TCP/IP Subagent supports both the deprecated IPv4-only MIB data and some of the version-neutral MIB data. Since IETF internet drafts expire in 6 months, copies of the IETF internet drafts mentioned in the bullets below are shipped with Communications Server and are installed in the z/OS UNIX file system in directory /usr/lpp/tcpip/samples with the following file names:

- ipmib.mi2 - IP-MIB
- ipfwdmib.mi2 - IP-FORWARD-MIB
- tcpmib.mi2 - TCP-MIB
- iaddrmib.mi2 - INET-ADDRESS-MIB from IETF draft-ietf-ops-rfc3291bis-01.txt. The INET-ADDRESS-MIB contains SNMP MIB data textual conventions referenced by MIB data definitions in the standard MIB modules.
- udpmib.mi2 - UDP-MIB

The INET-ADDRESS-MIB defines the types of IP addresses supported in version-neutral SNMP MIB data. The TCP/IP Subagent supports only the following types of IP addresses from the INET-ADDRESS-MIB:

- 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.

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 IETF Internet draft `draft-ietf-ipv6-rfc2011-update-04.txt`, and some additional IP counters from the Enterprise-specific MIB. The Subagent supports Route MIB data from the IP-FORWARD-MIB in Internet draft `draft-ietf-ipv6-rfc2096-update-05.txt`, 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 Interfaces Group MIB Using SMIv2*). 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
- TCP MIB data

  The Subagent supports the TCP MIB data from the TCP-MIB in IETF Internet draft `draft-ietf-ipv6-rfc2012-update-04.txt`. 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 IETF internet draft draft-ietf-ipv6-rfc2013-update-03.txt. 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, 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.

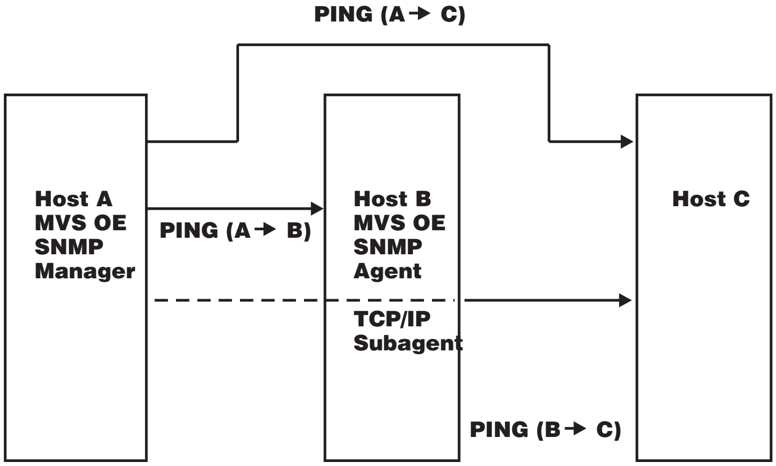


Figure 5. SNMP remote PING function

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.

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 only supports 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─┘
```

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 765.

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.

Example

The following is an example of using 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 is an example of using 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 18 for a description of what the variable value can represent.

Table 18. 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)

Table 18. SNMP Get command responses for variable value (continued)

Returned value	Description	Condition	Valid input
-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 implemented support for Interface MIB data from the IF-MIB from RFC 2233 (*The Interfaces Group MIB Using SMIv2*). For more information, see Appendix A, “SNMP capability statement,” on page 815 for a list of supported IF-MIB objects. RFC 2233 (*The Interfaces Group MIB Using SMIv2*) 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, where the following DEVICE, LINK, and INTERFACE profile statements are specified in the TCP/IP Profile data set:

```
DEVICE OSA1 ATM PORTNAME ATMPORT1
LINK ATMLINK1 ATM OSA1
DEVICE LCS1 LCS 100
LINK ETH1 ETHERNET 0 LCS1
INTERFACE GBIT1 DEFINE IPAQENET6 PORTNAME OSAQDIOG
IPADDR 2001:0DB8:0:1:0009:0067:0115:0066
```

The following interface entries would be created in the ifTable and ifXTable:

Table 19. Entries created in the ifTable and ifXTable

ifIndex	ifType	Description
1	53 (propVirtual)	LOOPBACK device
2	24 (softwareLOOPBACK)	LOOPBACK link
3	24 (softwareLOOPBACK)	LOOPBACK6 interface
4	53 (propVirtual)	ATM Device OSA1
5	49 (aal5)	ATMPORT1’s aal5 layer
6	37 (atm)	ATMPORT1’s atm layer
7	59/60 (atmlane8023/8025)	ATM LAN Emulation
8	59/60 (atmlane8023/8025)	ATM LAN Emulation
9	1 (other)	ATM Link ATMLINK1
10	53 (propVirtual)	LCS Device LCS1
11	6 (ethernetCsmacd)	LCS Link ETH1
12	53 (propVirtual)	MPCIPA Device OSAQDIOG
13	6 (ethernetCsmacd)	IPAQENET6 Interface GBIT1

The ifType values are assigned by the Internet Assigned Numbers Authority (IANA) to indicate the type of interface. In z/OS Communications Server, a DEVICE has a corresponding interface entry with its LINKs defined as interface entries stacked below it. An INTERFACE profile statement causes a device entry to be dynamically created in the interface tables, if no device has been defined. The ifStackTable is used to reflect the relationship between interfaces. A device is stacked above its links/interfaces. Its ifEntry and ifXEntry counters reflect the sum of its links/interfaces. See the IF-MIB for a detailed explanation of how the ifStackTable is used to reflect interface relationships.

In the previous example, a LOOPBACK device and LINK ifEntry were created when the links were not explicitly defined. TCP/IP automatically generated these entries. Also in this example, since the TCP/IP stack was enabled for IPv6 support, a LOOPBACK6 INTERFACE ifEntry was also automatically generated.

The ifEntry and ifXEntry counters can be retrieved from either the ETH1 LINK or LCS DEVICE interface entry to determine ETH1 activity. When there is only one link/interface defined for the device, the counters shown for a device interface entry equal those of a subordinate link/interface. When there are multiple links/interfaces defined for a device, the device counters are the sum of all the link/interface counters.

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 these ATM subordinate layers is obtained directly from OSA/SF. See “ATM-specific management data” on page 805 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 839 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 881 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 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 and OSA-Express2 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 *System z9 and zSeries 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 795 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 891 for information about RFCs.

The following MIB tables describe the supported OSA adapter 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 *System z9 and zSeries OSA-Express Customer's Guide and Reference* for instructions on obtaining a copy of the IBM-OSA-MIB module.

- **osaexpChannelTable**

  An entry in this table is created for every OSA-Express Ethernet or ATM feature and every OSA-Express2 Gigabit feature defined to TCP/IP by a DEVICE profile statement. The table contains descriptive and performance data. The values are retrieved from OSA/SF. This table is indexed by the ifIndex of the DEVICE 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 feature and every OSA-Express2 Gigabit feature defined to TCP/IP by a DEVICE profile statement, 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 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 feature and every OSA-Express2 Gigabit feature defined to TCP/IP by a LINK profile statement. 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 LINK and is defined in the IBM MVS Enterprise-specific MIB.
- **osaexpEthSnaTable**

  An entry in this table is created for every OSA-Express fast Ethernet adapter configured for SNA and defined to TCP/IP by a LINK profile statement. The values are retrieved from OSA/SF. This table is indexed by the ifIndex of the LINK and is defined in the IBM MVS Enterprise-specific MIB.
- **Interface Table Data**

  An entry is created in the ifTable and ifXTable table for every DEVICE and LINK profile statement which represents an OSA adapter. The ifTable and ifXTable data for OSA adapter DEVICE and LINK interfaces used by TCP/IP for data transport is retrieved from TCP/IP. Interface Table Data is defined in the ifMIB - RFC 2233 (*The Interfaces Group MIB Using SMIv2*)).
- **dot3StatsTable**

  An entry in this Ethernet table is created for every OSA-Express Ethernet or OSA-Express2 Gigabit feature defined to TCP/IP by an MPCIPA DEVICE profile statement. The values are retrieved from OSA/SF. This table 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 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 *System z9 and zSeries 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 (*The Interfaces Group MIB Using SMIv2*).
- **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 764.

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 only supported 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 only supported 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 839 for a list of all the DVIPA MIB objects. See "SNMP Enterprise-specific trap types" on page 884 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 will be an entry in this table for every connection being routed through the distributor. On a stack taking over a dynamic VIPA, there will be an entry in this table for every connection to the dynamic VIPA. On a sysplex distributor target stack or a stack which is in the process of giving up a dynamic VIPA, there will be 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 which are distributing connection requests as part of the sysplex distributor function. This table is not supported by stacks which are only targets 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 839 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 information about the data defined in the IBM MVS Enterprise-specific TN3270 MIB, and about how to cause connections to be monitored, see the Accessing remote hosts using Telnet information 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 will contain 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*.

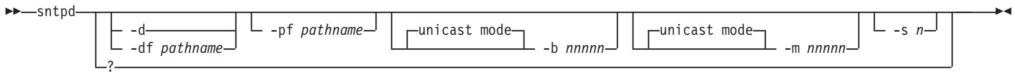
The z/OS UNIX sntpd command—Simple Network Time Protocol

Purpose

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 pid file. For example:

```
 -pf /var
```

-b *nnnnn*
Act in broadcast mode. Send local broadcasts on all interfaces every *nnnnn* seconds. The valid range for this value is 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. The valid range for this value is 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. The valid range for *n* is 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 one.

Note: The SNTP server will always respond 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
```

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.

```
   --
   --
   -- Program name : IBM z/OS Communications Server
   --                Capabilities ASN.1 Description file
   -- Requires:      IBM z/OS Communications Server
   --                Version 1 Release 9
   -- 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 "200606280000Z"
         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 capabilities
              statements.
              Licensed Materials - Property of IBM
              Restricted Materials of IBM
|             5694-A01 (C) Copyright IBM Corp. 1997, 2007
```

Figure 6. SNMP capability statement (Part 1 of 23)

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

Figure 6. SNMP capability statement (Part 2 of 23)

```
                    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
```

Figure 6. SNMP capability statement (Part 3 of 23)

```
             - 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
```

Figure 6. SNMP capability statement (Part 4 of 23)

```
             "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
```

Figure 6. SNMP capability statement (Part 5 of 23)

```
             - 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 1 Release 9 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.
```

Figure 6. SNMP capability statement (Part 6 of 23)

```
      -- 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"
      ::= { ibmTcpIpMvsCaps 1 }
   ibmTcpIpMvsDpiSaCaps AGENT-CAPABILITIES
      PRODUCT-RELEASE  "IBM z/OS Communications Server
|                       Version 1 Release 9 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,
                      ibmTCPIPmvsSystemGroup11,
                      ibmTCPIPmvsTcpGroup10,
                      ibmTCPIPmvsTcpGroupOld,
                      ibmTCPIPmvsUdpGroup3,
                      ibmTCPIPmvsUdpGroup4,
                      ibmTCPIPmvsInterfacesGroup9,
                      ibmTCPIPmvsPortGroup4,
                      ibmTCPIPmvsPortGroupOld,
|                     ibmTCPIPmvsRoutingGroup3,
                      ibmTCPIPmvsRoutingGroup,
                      ibmTCPIPmvsIpGroup,
                      ibmTCPIPmvsAtmSupportGroup4,
                      ibmTCPIPmvsAtmNotificationGroup,
```

Figure 6. SNMP capability statement (Part 7 of 23)

```
                      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        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
```

Figure 6. SNMP capability statement (Part 8 of 23)

```
            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        ifPhysAddress
            DESCRIPTION  "Only supported for the following interface
                          types when the interface is active:
                           - ATM
                           - HCH
                           - LCS Ethernet, Token Ring, FDDI
                           - MPCIPA Ethernet, Token Ring"
        VARIATION        ifAdminStatus
            SYNTAX       INTEGER { up(1), down(2) }
            DESCRIPTION  "Test mode (testing(3)) not supported. The
                          set operation is not allowed to a loopback
                          or Virtual IP Address (VIPA) interface.
                          This object reflects the desired state of
                          an interface. If a START command has been
```

Figure 6. SNMP capability statement (Part 9 of 23)

```
                  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)."
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        ifLastChange
    DESCRIPTION  "Use time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents time relative to the
                  agents IPL not TCP/IPs."
VARIATION        ifPromiscuousMode
    ACCESS        read-only
    DESCRIPTION  "Write access is not required, nor supported."
VARIATION        ifStackStatus
    SYNTAX       INTEGER { active(1) } -- subset of RowStatus
    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"
VARIATION        ifCounterDiscontinuityTime
    DESCRIPTION  "Use time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents time relative to the
                  agents IPL not TCP/IPs.  This value is
                  set only when an existing interface is
                  deleted from and then defined again to
                  the stack, or when certain errors occur
                  on an interface."
VARIATION        ifInNUcastPkts
    DESCRIPTION  "This implementation does not maintain this
                  object.  The value of the object will
                  always be zero."
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."
```

Figure 6. SNMP capability statement (Part 10 of 23)

```
        VARIATION        ifInBroadcastPkts
            DESCRIPTION  "Only supported for the following interface
                          types:
                           - LCS Ethernet, Token Ring, FDDI
                           - MPCIPA Ethernet, Token Ring,
                                    HiperSockets"
        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"
    SUPPORTS             IP-MIB   -- from IETF internet draft
                         -- draft-ietf-ipv6-rfc2011-update-04.txt
        INCLUDES         { ipGroup, icmpGroup,
                           ipSystemStatsGroup,   ipAddressGroup,
                           ipNetToMediaGroup,    ipDefaultRouterGroup,
                           icmpGroup2,
                           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
```

Figure 6. SNMP capability statement (Part 11 of 23)

```
                  because the index is an IP address and
                  the TCP/IP stack allows the same IP
                  address to be defined for multiple
                  interfaces."
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        ip6Forwarding
    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        ip6DefaultHopLimit
    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        ipv6IfTableLastChange
    DESCRIPTION  "Uses time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents time relative to the
                  agents IPL not TCP/IPs."
VARIATION        ipv6InterfaceReasmMaxSize
    DESCRIPTION  "The value of this MIB object will always
                  be 65535."
VARIATION        ipv6InterfaceAdminStatus
    ACCESS        read-only
    DESCRIPTION  "The value of this MIB object will always
                  be up(1).  Write access is not supported."
VARIATION        ipSystemStatsDiscontinuityTime
    DESCRIPTION  "Uses time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents time relative to the
```

Figure 6. SNMP capability statement (Part 12 of 23)

```
                  agents IPL not TCP/IPs.  This value is
                  set only when an existing interface is
                  deleted from and then defined again to
                  the stack, or 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        ipAddressCreated
    DESCRIPTION  "Uses time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents 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 respesents time relative to the
                  agents IPL not TCP/IPs."
VARIATION        inetNetToMediaLastUpdated
    DESCRIPTION  "Uses time that TCP/IP was started instead of
                  sysUpTime to calculate this value, since
                  sysUpTime respesents 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        ipDefaultRouterPreference
    DESCRIPTION  "This implementation does not support IETF
                  draft-ietf-ipv6-router-selection-02.txt,
                  so the value of this MIB object will always
                  be medium(0)."
VARIATION        ipLastChangeGroup
```

Figure 6. SNMP capability statement (Part 13 of 23)

```
            DESCRIPTION  "This implementation only supports the
                          ipv6IfTableLastChange object from this
                          group."
    SUPPORTS             IP-FORWARD-MIB   -- from IETF internet draft
                         -- draft-ietf-ipv6-rfc2096-update-05.txt
        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        inetCidrRouteStatus
```

Figure 6. SNMP capability statement (Part 14 of 23)

```
        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   -- from IETF internet draft
                     -- draft-ietf-ipv6-rfc2012-update-04.txt
    INCLUDES         { tcpGroup,
                       tcpBaseGroup, tcpConnectionGroup,
                       tcpListenerGroup,
                       tcpHCGroup}
    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   -- from IETF internet draft
                     -- draft-ietf-ipv6-rfc2013-update-03.txt
    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."
```

Figure 6. SNMP capability statement (Part 15 of 23)

```
    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        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
```

Figure 6. SNMP capability statement (Part 16 of 23)

```
                      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
```

Figure 6. SNMP capability statement (Part 17 of 23)

```
                       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     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 }
```

Figure 6. SNMP capability statement (Part 18 of 23)

```
    ibmTcpIpMvsOspfCaps AGENT-CAPABILITIES
       PRODUCT-RELEASE  "IBM z/OS Communications Server
|                        Version 1 Release 9 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
```

Figure 6. SNMP capability statement (Part 19 of 23)

```
        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
```

Figure 6. SNMP capability statement (Part 20 of 23)

```
    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
```

Figure 6. SNMP capability statement (Part 21 of 23)

```
    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."
```

Figure 6. SNMP capability statement (Part 22 of 23)

```
        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 1 Release 9 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 1 Release 9 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
```

Figure 6. SNMP capability statement (Part 23 of 23)

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 20 on page 841 shows the MIB objects supported by z/OS Communications Server IP SNMP agent and subagents.

Table 20. 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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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	RFC2011	R/W
ipDefaultTTL	1.3.6.1.2.1.4.2	TCP/IP	RFC2011	R/W
ipInReceives	1.3.6.1.2.1.4.3	TCP/IP	RFC2011	R/O
ipInHdrErrors	1.3.6.1.2.1.4.4	TCP/IP	RFC2011	R/O
ipInAddrErrors	1.3.6.1.2.1.4.5	TCP/IP	RFC2011	R/O
ipForwDatagrams	1.3.6.1.2.1.4.6	TCP/IP	RFC2011	R/O
ipInUnknownProtos	1.3.6.1.2.1.4.7	TCP/IP	RFC2011	R/O
ipInDiscards	1.3.6.1.2.1.4.8	TCP/IP	RFC2011	R/O
ipInDelivers	1.3.6.1.2.1.4.9	TCP/IP	RFC2011	R/O
ipOutRequests	1.3.6.1.2.1.4.10	TCP/IP	RFC2011	R/O
ipOutDiscards	1.3.6.1.2.1.4.11	TCP/IP	RFC2011	R/O
ipOutNoRoutes	1.3.6.1.2.1.4.12	TCP/IP	RFC2011	R/O
ipReasmTimeout	1.3.6.1.2.1.4.13	TCP/IP	RFC2011	R/W
ipReasmReqds	1.3.6.1.2.1.4.14	TCP/IP	RFC2011	R/O
ipReasmOKs	1.3.6.1.2.1.4.15	TCP/IP	RFC2011	R/O
ipReasmFails	1.3.6.1.2.1.4.16	TCP/IP	RFC2011	R/O
ipFragOKs	1.3.6.1.2.1.4.17	TCP/IP	RFC2011	R/O
ipFragFails	1.3.6.1.2.1.4.18	TCP/IP	RFC2011	R/O
ipFragCreates	1.3.6.1.2.1.4.19	TCP/IP	RFC2011	R/O
ipAddrTable	1.3.6.1.2.1.4.20	TCP/IP	RFC2011	N/A
ipAddrEntry	1.3.6.1.2.1.4.20.1	TCP/IP	RFC2011	N/A
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	TCP/IP	RFC2011	R/O
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	TCP/IP	RFC2011	R/O
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	TCP/IP	RFC2011	R/O
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	TCP/IP	RFC2011	R/O
ipAdEntReasmMaxSize	1.3.6.1.2.1.4.20.1.5	TCP/IP	RFC2011	R/O
ipNetToMediaTable	1.3.6.1.2.1.4.22	TCP/IP	RFC2011	N/A
ipNetToMediaEntry	1.3.6.1.2.1.4.22.1	TCP/IP	RFC2011	N/A
ipNetToMediaIfIndex	1.3.6.1.2.1.4.22.1.1	TCP/IP	RFC2011	R/O
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	TCP/IP	RFC2011	R/O
ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3	TCP/IP	RFC2011	R/O
ipNetToMediaType	1.3.6.1.2.1.4.22.1.4	TCP/IP	RFC2011	R/O
ipRoutingDiscards	1.3.6.1.2.1.4.23	TCP/IP	RFC2011	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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	draft-RFC2096	R/O
inetCidrRouteTable	1.3.6.1.2.1.4.24.7	TCP/IP	draft-RFC2096	N/A
inetCidrRouteEntry	1.3.6.1.2.1.4.24.7.1	TCP/IP	draft-RFC2096	N/A
inetCidrRouteDestType	1.3.6.1.2.1.4.24.7.1.1	TCP/IP	draft-RFC2096	N/A
inetCidrRouteDest	1.3.6.1.2.1.4.24.7.1.2	TCP/IP	draft-RFC2096	N/A
inetCidrRoutePfxLen	1.3.6.1.2.1.4.24.7.1.3	TCP/IP	draft-RFC2096	N/A
inetCidrRoutePolicy	1.3.6.1.2.1.4.24.7.1.4	TCP/IP	draft-RF C2096	N/A
inetCidrRouteNextHopType	1.3.6.1.2.1.4.24.7.1.5	TCP/IP	draft-RFC2096	N/A
inetCidrRouteNextHop	1.3.6.1.2.1.4.24.7.1.6	TCP/IP	draft-RFC2096	N/A
inetCidrRouteIfIndex	1.3.6.1.2.1.4.24.7.1.7	TCP/IP	draft-RFC2096	R/O
inetCidrRouteType	1.3.6.1.2.1.4.24.7.1.8	TCP/IP	draft-RFC2096	R/O
inetCidrRouteProto	1.3.6.1.2.1.4.24.7.1.9	TCP/IP	draft-RFC2096	R/O
inetCidrRouteAge	1.3.6.1.2.1.4.24.7.1.10	TCP/IP	draft-RFC2096	R/O
inetCidrRouteNextHopAS	1.3.6.1.2.1.4.24.7.1.11	TCP/IP	draft-RFC2096	R/O
inetCidrRouteMetric1	1.3.6.1.2.1.4.24.7.1.12	TCP/IP	draft-RFC2096	R/O
inetCidrRouteMetric2	1.3.6.1.2.1.4.24.7.1.13	TCP/IP	draft-RFC2096	R/O
inetCidrRouteMetric3	1.3.6.1.2.1.4.24.7.1.14	TCP/IP	draft-RFC2096	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
inetCidrRouteMetric4	1.3.6.1.2.1.4.24.7.1.15	TCP/IP	draft-RFC2096	R/O
inetCidrRouteMetric5	1.3.6.1.2.1.4.24.7.1.16	TCP/IP	draft-RFC2096	R/O
inetCidrRouteStatus	1.3.6.1.2.1.4.24.7.1.17	TCP/IP	draft-RFC2096	R/O
inetCidrRouteDiscards	1.3.6.1.2.1.4.24.8	TCP/IP	draft-RFC2096	R/O
ip6Forwarding	1.3.6.1.2.1.4.25	TCP/IP	draft-RFC2011	R/W
ip6DefaultHopLimit	1.3.6.1.2.1.4.26	TCP/IP	draft-RF C2011	R/W
ipv6IfTableLastChange	1.3.6.1.2.1.4.29	TCP/IP	draft-RFC2011	R/O
ipv6InterfaceTable	1.3.6.1.2.1.4.30	TCP/IP	draft-RFC2011	N/A
ipv6InterfaceEntry	1.3.6.1.2.1.4.30.1	TCP/IP	draft-RFC2011	N/A
ipv6InterfaceIfIndex	1.3.6.1.2.1.4.30.1.1	TCP/IP	draft-RFC2011	N/A
ipv6InterfaceReasmMaxSize	1.3.6.1.2.1.4.30.1.2	TCP/IP	draft-RFC2011	R/O
ipv6InterfaceIdentifier	1.3.6.1.2.1.4.30.1.3	TCP/IP	draft-RFC2011	R/O
ipv6InterfacePhysicalAddress	1.3.6.1.2.1.4.30.1.4	TCP/IP	draft-RFC2011	R/O
ipv6InterfaceAdminStatus	1.3.6.1.2.1.4.30.1.5	TCP/IP	draft-RFC2011	R/O
ipv6InterfaceReachableTime	1.3.6.1.2.1.4.30.1.6	TCP/IP	draft-RFC2011	R/O
ipv6InterfaceRetransmitTime	1.3.6.1.2.1.4.30.1.7	TCP/IP	draft-RFC2011	R/O
ipTrafficStats	1.3.6.1.2.1.4.31	TCP/IP	draft-RFC2011	N/A
ipSystemStatsTable	1.3.6.1.2.1.4.31.1	TCP/IP	draft-RFC2011	N/A
ipSystemStatsEntry	1.3.6.1.2.1.4.31.1.1	TCP/IP	draft-RFC2011	N/A
ipSystemStatsAFType	1.3.6.1.2.1.4.31.1.1.1	TCP/IP	draft-RFC2011	N/A
ipSystemStatsInReceives	1.3.6.1.2.1.4.31.1.1.3	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCInReceives	1.3.6.1.2.1.4.31.1.1.4	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInOctets	1.3.6.1.2.1.4.31.1.1.5	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCInOctets	1.3.6.1.2.1.4.31.1.1.6	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInHdrErrors	1.3.6.1.2.1.4.31.1.1.7	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInNoRoutes	1.3.6.1.2.1.4.31.1.1.8	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInAddrErrors	1.3.6.1.2.1.4.31.1.1.9	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInUnknownProtos	1.3.6.1.2.1.4.31.1.1.10	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInTruncatedPkts	1.3.6.1.2.1.4.31.1.1.11	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInForwDatagrams	1.3.6.1.2.1.4.31.1.1.12	TCP/IP	draft-RFC2011	R/O
ipSystemStatsReasmReqds	1.3.6.1.2.1.4.31.1.1.13	TCP/IP	draft-RFC2011	R/O
ipSystemStatsReasmOKs	1.3.6.1.2.1.4.31.1.1.14	TCP/IP	draft-RFC2011	R/O
ipSystemStatsReasmFails	1.3.6.1.2.1.4.31.1.1.15	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInDiscards	1.3.6.1.2.1.4.31.1.1.16	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInDelivers	1.3.6.1.2.1.4.31.1.1.17	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutRequests	1.3.6.1.2.1.4.31.1.1.18	TCP/IP	draft-RFC2011	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipSystemStatsOutNoRoutes	1.3.6.1.2.1.4.31.1.1.19	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutForwDatagrams	1.3.6.1.2.1.4.31.1.1.20	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutDiscards	1.3.6.1.2.1.4.31.1.1.21	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutFragReqds	1.3.6.1.2.1.4.31.1.1.22	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutFragOKs	1.3.6.1.2.1.4.31.1.1.23	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutFragFails	1.3.6.1.2.1.4.31.1.1.24	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutFragCreates	1.3.6.1.2.1.4.31.1.1.25	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutTransmits	1.3.6.1.2.1.4.31.1.1.26	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCOutTransmits	1.3.6.1.2.1.4.31.1.1.27	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutOctets	1.3.6.1.2.1.4.31.1.1.28	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCOutOctets	1.3.6.1.2.1.4.31.1.1.29	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInMcastPkts	1.3.6.1.2.1.4.31.1.1.30	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCInMcastPkts	1.3.6.1.2.1.4.31.1.1.31	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInMcastOctets	1.3.6.1.2.1.4.31.1.1.32	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCInMcastOctets	1.3.6.1.2.1.4.31.1.1.33	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutMcastPkts	1.3.6.1.2.1.4.31.1.1.34	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCOutMcastPkts	1.3.6.1.2.1.4.31.1.1.35	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutMcastOctets	1.3.6.1.2.1.4.31.1.1.36	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCOutMcastOctets	1.3.6.1.2.1.4.31.1.1.37	TCP/IP	draft-RFC2011	R/O
ipSystemStatsInBcastPkts	1.3.6.1.2.1.4.31.1.1.38	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCInBcastPkts	1.3.6.1.2.1.4.31.1.1.39	TCP/IP	draft-RFC2011	R/O
ipSystemStatsOutBcastPkts	1.3.6.1.2.1.4.31.1.1.40	TCP/IP	draft-RFC2011	R/O
ipSystemStatsHCOutBcastPkts	1.3.6.1.2.1.4.31.1.1.41	TCP/IP	draft-RFC2011	R/O
ipSystemStatsDiscontinuityTime	1.3.6.1.2.1.4.31.1.1.42	TCP/IP	draft-RFC2011	R/O
ipSystemStatsRefreshRate	1.3.6.1.2.1.4.31.1.1.43	TCP/IP	draft-RFC2011	R/O
ipAddressPrefixTable	1.3.6.1.2.1.4.32	TCP/IP	draft-RFC2011	N/A
ipAddressPrefixEntry	1.3.6.1.2.1.4.32.1	TCP/IP	draft-RFC2011	N/A
ipAddressPrefixIfIndex	1.3.6.1.2.1.4.32.1.1	TCP/IP	draft-RFC2011	N/A
ipAddressPrefixType	1.3.6.1.2.1.4.32.1.2	TCP/IP	draft-RFC2011	N/A
ipAddressPrefixPrefix	1.3.6.1.2.1.4.32.1.3	TCP/IP	draft_RFC2011	N/A
ipAddressPrefixLength	1.3.6.1.2.1.4.32.1.4	TCP/IP	draft_RFC2011	N/A
ipAddressPrefixOrigin	1.3.6.1.2.1.4.32.1.5	TCP/IP	draft_RFC2011	R/O
ipAddressPrefixOnLinkFlag	1.3.6.1.2.1.4.32.1.6	TCP/IP	draft_RFC2011	R/O
ipAddressPrefixAutonomousFlag	1.3.6.1.2.1.4.32.1.7	TCP/IP	draft_RFC2011	R/O
ipAddressPrefixAdvPreferredLifetime	1.3.6.1.2.1.4.32.1.8	TCP/IP	draft_RFC2011	R/O
ipAddressPrefixAdvValidLifetime	1.3.6.1.2.1.4.32.1.9	TCP/IP	draft_RFC2011	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipAddressTable	1.3.6.1.2.1.4.33	TCP/IP	draft-RFC2011	N/A
ipAddressEntry	1.3.6.1.2.1.4.33.1	TCP/IP	draft-RFC2011	N/A
ipAddressAddrType	1.3.6.1.2.1.4.33.1.1	TCP/IP	draft-RFC2011	N/A
ipAddressAddr	1.3.6.1.2.1.4.33.1.2	TCP/IP	draft-RFC2011	N/A
ipAddressIfIndex	1.3.6.1.2.1.4.33.1.3	TCP/IP	draft-RFC2011	R/O
ipAddressType	1.3.6.1.2.1.4.33.1.4	TCP/IP	draft-RFC2011	R/O
ipAddressPrefix	1.3.6.1.2.1.4.33.1.5	TCP/IP	draft-RFC2011	R/O
ipAddressOrigin	1.3.6.1.2.1.4.33.1.6	TCP/IP	draft-RFC2011	R/O
ipAddressStatus	1.3.6.1.2.1.4.33.1.7	TCP/IP	draft-RFC2011	R/O
ipAddressCreated	1.3.6.1.2.1.4.33.1.8	TCP/IP	draft-RFC2011	R/O
ipAddressLastChanged	1.3.6.1.2.1.4.33.1.9	TCP/IP	draft-RFC2011	R/O
inetNetToMediaTable	1.3.6.1.2.1.4.34	TCP/IP	draft-RFC2011	N/A
inetNetToMediaEntry	1.3.6.1.2.1.4.34.1	TCP/IP	draft-RFC2011	N/A
inetNetToMediaIfIndex	1.3.6.1.2.1.4.34.1.1	TCP/IP	draft-RFC2011	N/A
inetNetToMediaNetAddressType	1.3.6.1.2.1.4.34.1.2	TCP/IP	draft-RFC2011	N/A
inetNetToMediaNetAddress	1.3.6.1.2.1.4.34.1.3	TCP/IP	draft-RFC2011	N/A
inetNetToMediaPhysAddress	1.3.6.1.2.1.4.34.1.4	TCP/IP	draft-RFC2011	R/O
inetNetToMediaLastUpdated	1.3.6.1.2.1.4.34.1.5	TCP/IP	draft-RFC2011	R/O
inetNetToMediaType	1.3.6.1.2.1.4.34.1.6	TCP/IP	draft-RFC2011	R/O
inetNetToMediaState	1.3.6.1.2.1.4.34.1.7	TCP/IP	draft-RFC2011	R/O
ipDefaultRouterTable	1.3.6.1.2.1.4.36	TCP/IP	draft-RFC2011	N/A
ipDefaultRouterEntry	1.3.6.1.2.1.4.36.1	TCP/IP	draft-RFC2011	N/A
ipDefaultRouterAFType	1.3.6.1.2.1.4.36.1.1	TCP/IP	draft-RFC2011	N/A
ipDefaultRouterAddress	1.3.6.1.2.1.4.36.1.2	TCP/IP	draft-RFC2011	N/A
ipDefaultRouterIfIndex	1.3.6.1.2.1.4.36.1.3	TCP/IP	draft-RFC2011	R/O
ipDefaultRouterLifetime	1.3.6.1.2.1.4.36.1.4	TCP/IP	draft-RFC2011	R/O
ipDefaultRouterPreference	1.3.6.1.2.1.4.36.1.5	TCP/IP	draft-RFC2011	R/O
icmpInMsgs	1.3.6.1.2.1.5.1	TCP/IP	RFC2011	R/O
icmpInErrors	1.3.6.1.2.1.5.2	TCP/IP	RFC2011	R/O
icmpInDestUnreachs	1.3.6.1.2.1.5.3	TCP/IP	RFC2011	R/O
icmpInTimeExcds	1.3.6.1.2.1.5.4	TCP/IP	RFC2011	R/O
icmpInParmProbs	1.3.6.1.2.1.5.5	TCP/IP	RFC2011	R/O
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	TCP/IP	RFC2011	R/O
icmpInRedirects	1.3.6.1.2.1.5.7	TCP/IP	RFC2011	R/O
icmpInEchos	1.3.6.1.2.1.5.8	TCP/IP	RFC2011	R/O
icmpInEchoReps	1.3.6.1.2.1.5.9	TCP/IP	RFC2011	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
icmpInTimestamps	1.3.6.1.2.1.5.10	TCP/IP	RFC2011	R/O
icmpInTimestampReps	1.3.6.1.2.1.5.11	TCP/IP	RFC2011	R/O
icmpInAddrMasks	1.3.6.1.2.1.5.12	TCP/IP	RFC2011	R/O
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	TCP/IP	RFC2011	R/O
icmpOutMsgs	1.3.6.1.2.1.5.14	TCP/IP	RFC2011	R/O
icmpOutErrors	1.3.6.1.2.1.5.15	TCP/IP	RFC2011	R/O
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	TCP/IP	RFC2011	R/O
icmpOutTimeExcds	1.3.6.1.2.1.5.17	TCP/IP	RFC2011	R/O
icmpOutParmProbs	1.3.6.1.2.1.5.18	TCP/IP	RFC2011	R/O
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	TCP/IP	RFC2011	R/O
icmpOutRedirects	1.3.6.1.2.1.5.20	TCP/IP	RFC2011	R/O
icmpOutEchos	1.3.6.1.2.1.5.21	TCP/IP	RFC2011	R/O
icmpOutEchoReps	1.3.6.1.2.1.5.22	TCP/IP	RFC2011	R/O
icmpOutTimestamps	1.3.6.1.2.1.5.23	TCP/IP	RFC2011	R/O
icmpOutTimestampReps	1.3.6.1.2.1.5.24	TCP/IP	RFC2011	R/O
icmpOutAddrMasks	1.3.6.1.2.1.5.25	TCP/IP	RFC2011	R/O
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	TCP/IP	RFC2011	R/O
inetIcmpTable	1.3.6.1.2.1.5.27	TCP/IP	draft-RFC2011	N/A
inetIcmpEntry	1.3.6.1.2.1.5.27.1	TCP/IP	draft-RFC2011	N/A
inetIcmpAFType	1.3.6.1.2.1.5.27.1.1	TCP/IP	draft-RFC2011	N/A
inetIcmpInMsgs	1.3.6.1.2.1.5.27.1.2	TCP/IP	draft-RFC2011	R/O
inetIcmpInErrors	1.3.6.1.2.1.5.27.1.3	TCP/IP	draft-RFC2011	R/O
inetIcmpOutMsgs	1.3.6.1.2.1.5.27.1.4	TCP/IP	draft-RFC2011	R/O
inetIcmpOutErrors	1.3.6.1.2.1.5.27.1.5	TCP/IP	draft-RFC2011	R/O
inetIcmpMsgTable	1.3.6.1.2.1.5.28	TCP/IP	draft-RFC2011	N/A
inetIcmpMsgEntry	1.3.6.1.2.1.5.28.1	TCP/IP	draft-RFC2011	N/A
inetIcmpMsgAFType	1.3.6.1.2.1.5.28.1.1	TCP/IP	draft-RFC2011	N/A
inetIcmpMsgType	1.3.6.1.2.1.5.28.1.2	TCP/IP	draft-RFC2011	N/A
inetIcmpMsgInPkts	1.3.6.1.2.1.5.28.1.3	TCP/IP	draft-RFC2011	R/O
inetIcmpMsgOutPkts	1.3.6.1.2.1.5.28.1.4	TCP/IP	draft-RFC2011	R/O
tcpRtoAlgorithm	1.3.6.1.2.1.6.1	TCP/IP	RFC2012	R/O
tcpRtoMin	1.3.6.1.2.1.6.2	TCP/IP	RFC2012	R/O
tcpRtoMax	1.3.6.1.2.1.6.3	TCP/IP	RFC2012	R/O
tcpMaxConn	1.3.6.1.2.1.6.4	TCP/IP	RFC2012	R/O
tcpActiveOpens	1.3.6.1.2.1.6.5	TCP/IP	RFC2012	R/O
tcpPassiveOpens	1.3.6.1.2.1.6.6	TCP/IP	RFC2012	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
tcpAttemptFails	1.3.6.1.2.1.6.7	TCP/IP	RFC2012	R/O
tcpEstabResets	1.3.6.1.2.1.6.8	TCP/IP	RFC2012	R/O
tcpCurrEstab	1.3.6.1.2.1.6.9	TCP/IP	RFC2012	R/O
tcpInSegs	1.3.6.1.2.1.6.10	TCP/IP	RFC2012	R/O
tcpOutSegs	1.3.6.1.2.1.6.11	TCP/IP	RFC2012	R/O
tcpRetransSegs	1.3.6.1.2.1.6.12	TCP/IP	RFC2012	R/O
tcpConnTable	1.3.6.1.2.1.6.13	TCP/IP	RFC2012	N/A
tcpConnEntry	1.3.6.1.2.1.6.13.1	TCP/IP	RFC2012	N/A
tcpConnState	1.3.6.1.2.1.6.13.1.1	TCP/IP	RFC2012	R/W
tcpConnLocalAddress	1.3.6.1.2.1.6.13.1.2	TCP/IP	RFC2012	R/O
tcpConnLocalPort	1.3.6.1.2.1.6.13.1.3	TCP/IP	RFC2012	R/O
tcpConnRemAddress	1.3.6.1.2.1.6.13.1.4	TCP/IP	RFC2012	R/O
tcpConnRemPort	1.3.6.1.2.1.6.13.1.5	TCP/IP	RFC2012	R/O
tcpInErrs	1.3.6.1.2.1.6.14	TCP/IP	RFC2012	R/O
tcpOutRsts	1.3.6.1.2.1.6.15	TCP/IP	RFC2012	R/O
tcpHCInSegs	1.3.6.1.2.1.6.17	TCP/IP	draft-RFC2012	R/O
tcpHCOutSegs	1.3.6.1.2.1.6.18	TCP/IP	draft-RFC2012	R/O
tcpConnectionTable	1.3.6.1.2.1.6.19	TCP/IP	draft-RFC2012	N/A
tcpConnectionEntry	1.3.6.1.2.1.6.19.1	TCP/IP	draft-RFC2012	N/A
tcpConnectionLocalAddressType	1.3.6.1.2.1.6.19.1.1	TCP/IP	draft-RFC2012	N/A
tcpConnectionLocalAddress	1.3.6.1.2.1.6.19.1.2	TCP/IP	draft-RFC2012	N/A
tcpConnectionLocalPort	1.3.6.1.2.1.6.19.1.3	TCP/IP	draft-RFC2012	N/A
tcpConnectionRemAddressType	1.3.6.1.2.1.6.19.1.4	TCP/IP	draft-RFC2012	N/A
tcpConnectionRemAddress	1.3.6.1.2.1.6.19.1.5	TCP/IP	draft-RFC2012	N/A
tcpConnectionRemPort	1.3.6.1.2.1.6.19.1.6	TCP/IP	draft-RFC2012	N/A
tcpConnectionState	1.3.6.1.2.1.6.19.1.7	TCP/IP	draft-RFC2012	R/W
tcpConnectionProcess	1.3.6.1.2.1.6.19.1.8	TCP/IP	draft-RFC2012	R/O
tcpListenerTable	1.3.6.1.2.1.6.20	TCP/IP	draft-RFC2012	N/A
tcpListenerEntry	1.3.6.1.2.1.6.20.1	TCP/IP	draft-RFC2012	N/A
tcpListenerLocalAddressType	1.3.6.1.2.1.6.20.1.1	TCP/IP	draft-RFC2012	N/A
tcpListenerLocalAddress	1.3.6.1.2.1.6.20.1.2	TCP/IP	draft-RFC2012	N/A
tcpListenerLocalPort	1.3.6.1.2.1.6.20.1.3	TCP/IP	draft-RFC2012	N/A
tcpListenerProcess	1.3.6.1.2.1.6.20.1.4	TCP/IP	draft-RFC2012	R/O
udpInDatagrams	1.3.6.1.2.1.7.1	TCP/IP	RFC2013	R/O
udpNoPorts	1.3.6.1.2.1.7.2	TCP/IP	RFC2013	R/O
udpInErrors	1.3.6.1.2.1.7.3	TCP/IP	RFC2013	R/O

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
udpOutDatagrams	1.3.6.1.2.1.7.4	TCP/IP	RFC2013	R/O
udpTable	1.3.6.1.2.1.7.5	TCP/IP	RFC2013	N/A
udpEntry	1.3.6.1.2.1.7.5.1	TCP/IP	RFC2013	N/A
udpLocalAddress	1.3.6.1.2.1.7.5.1.1	TCP/IP	RFC2013	R/O
udpLocalPort	1.3.6.1.2.1.7.5.1.2	TCP/IP	RFC2013	R/O
udpEndpointTable	1.3.6.1.2.1.7.7	TCP/IP	draft-RFC2013	N/A
udpEndpointEntry	1.3.6.1.2.1.7.7.1	TCP/IP	draft-RFC2013	N/A
udpEndpointLocalAddressType	1.3.6.1.2.1.7.7.1.1	TCP/IP	draft-RFC2013	N/A
udpEndpointLocalAddress	1.3.6.1.2.1.7.7.1.2	TCP/IP	draft-RFC2013	N/A
udpEndpointLocalPort	1.3.6.1.2.1.7.7.1.3	TCP/IP	draft-RFC2013	N/A
udpEndpointRemoteAddressType	1.3.6.1.2.1.7.7.1.4	TCP/IP	draft-RFC2013	N/A
udpEndpointRemoteAddress	1.3.6.1.2.1.7.7.1.5	TCP/IP	draft-RFC2013	N/A
udpEndpointRemotePort	1.3.6.1.2.1.7.7.1.6	TCP/IP	draft-RFC2013	N/A
udpEndpointInstance	1.3.6.1.2.1.7.7.1.7	TCP/IP	draft-RFC2013	R/O
udpEndpointProcess	1.3.6.1.2.1.7.7.1.8	TCP/IP	draft-RFC2013	R/O
udpHCInDatagrams	1.3.6.1.2.1.7.8	TCP/IP	draft-RFC2013	R/O
udpHCOutDatagrams	1.3.6.1.2.1.7.9	TCP/IP	draft-RFC2013	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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ipoaVcType	1.3.6.1.2.1.10.46.1.7.1.3	TCP/IP	RFC2320	R/O
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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 20. 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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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 20. 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
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
ibmMvsGatewayVarianceGain	1.3.6.1.4.1.2.6.19.2.2.5.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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
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

Table 20. MIB objects (continued)

Object descriptor	Object identifier	Supported by	Defined by	Access allowed
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 21. 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 21. 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 22 lists the generic trap types that can be received by SNMP.

Table 22. 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.

Table 22. Generic trap types (continued)

Value	Type	Description
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 23 lists the Enterprise-specific trap types generated by subagents shipped with z/OS Communications Server. All Enterprise-specific traps are generated with a trap value of 6.

Table 23. 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.

Table 23. MVS Enterprise trap types (continued)

Value	Subagent	Type	Description
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 will indicate the dynamic VIPA IP address, distributed port, and target stack dynamic XCF IP address.
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 will indicate 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 will indicate 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 will indicate 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.

Table 23. 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 23. 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

Table 24 lists the Internet Control Message Protocol (ICMP) types and codes from *TCP/IP Illustrated, Volume 1 The Protocols*, by W. Richard Stevens.

Table 24. ICMP types and codes

Type	Code	Description
0	0	echo reply
3	destination unreachable	
	0	network unreachable
	1	host unreachable
	2	protocol unreachable
	3	port unreachable
	4	fragmentation needed
	5	source route failed
	6	destination network unknown
	7	destination host unknown
	8	source host isolated
	9	destination network administratively prohibited
	10	destination host administratively prohibited
	11	network unreachable for ToS
	12	host unreachable for ToS
	13	communication administratively prohibited by filtering
	14	host precedence violation
	15	precedence cutoff in effect
4	0	source quench
5	redirect	
	0	redirect for network
	1	redirect for host
	2	redirect for type of service and network
	3	redirect for type of service and host
8	0	echo request
9	0	router advertisement
10	0	router solicitation
11	time exceeded	
	0	time-to-live equals 0 during transmit
	1	time-to-live equals 0 during reassembly
12	parameter problem	
	0	IP header bad
	1	required option missing
13	0	timestamp request

Table 24. ICMP types and codes (continued)

Type	Code	Description
14	0	timestamp reply
15	0	information request
16	0	information reply
17	0	address mask request
18	0	address mask reply

Table 25 lists the Internet Control Message Protocol for IPv6 (ICMPv6) types and codes.

Table 25. ICMPv6 types and codes

Type	Code	Description
1	destination unreachable	
	0	no route to destination
	1	communication with destination administratively prohibited
	2	beyond scope of source address
	3	address unreachable
	4	port unreachable
2	0	packet too big
3	time exceeded	
	0	hop limit exceeded in transit
	1	fragment reassembly time exceeded
4	parameter problem	
	0	erroneous header field encountered
	1	unrecognized Next Header type encountered
	2	unrecognized IPv6 option encountered
128	0	echo request
129	0	echo reply
130	0	group membership query
131	0	group membership reply
132	0	group membership reduction
133	0	router solicitation
134	0	router advertisement
135	0	neighbor solicitation
136	0	neighbor advertisement
137	0	redirect

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.

See "Internet drafts" on page 906 for draft RFCs implemented in this and previous Communications Server releases.

Many features of TCP/IP Services are based on the following RFCs:

RFC	**Title and Author**
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 951 *Bootstrap Protocol* W.J. Croft, J. Gilmore

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 1497 *BOOTP Vendor Information Extensions* J. Reynolds

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 1533 *DHCP Options and BOOTP Vendor Extensions* S. Alexander, R. Droms

RFC 1534 *Interoperation Between DHCP and BOOTP* R. Droms

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 1541 *Dynamic Host Configuration Protocol* R. Droms

RFC 1542 *Clarifications and Extensions for the Bootstrap Protocol* W. Wimer

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 2132 *DHCP Options and BOOTP Vendor Extensions* S. Alexander, R. Droms

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 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 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 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 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 4007 *IPv6 Scoped Address Architecture* S. Deering, B. Haberman, T. Jinmei, E. Nordmark, B. Zill

RFC 4217 *Securing FTP with TLS* P. Ford-Hutchinson

Internet drafts

Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Other groups may also distribute working documents as Internet drafts. You can see Internet drafts at http://www.ietf.org/ID.html.

Several areas of IPv6 implementation include elements of the following Internet drafts and are subject to change during the RFC review process.

Draft **Title and Author**

draft-bivens-sasp-02
Server/Application State Protocol v1 A. Bivens

draft-ietf-ipngwg-icmp-v3-07
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

draft-ietf-ipsec-esp-v3-10
IP Encapsulating Security Payload (ESP) S. Kent

draft-ietf-ipsec-rfc2402bis-11
IP Authentication Header S. Kent

draft-ietf-ipsec-rfc2401bis-06
Security Architecture for the Internet Protocol S. Kent, K. Seo

draft-ietf-ospf-ospfv3-auth-07
Authentication/Confidentiality for OSPFv3 M. Gupta, N. Melam

Appendix G. Information APARs and technotes

This appendix lists information APARs for IP and SNA documents.

Note:

1. Information APARs contain updates to previous editions of the documents listed in Table 26 and Table 27 on page 908. Documents updated for V1R9 are complete except for the updates contained in the information APARs that might be issued after V1R9 documents went to press.
2. Information APARs are predefined for z/OS V1R9 Communications Server and might not contain updates.
3. Information APARs for z/OS documents are in the document called *z/OS and z/OS.e DOC APAR and PTF ++HOLD Documentation*, which can be found at http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ZIDOCMST/CCONTENTS.

Information APARs for IP documents

Table 26 lists information APARs for V1R6 IP documents. For releases V1R7 and later, updates are available as technotes, which can be found at http://www.ibm.com/support/docview.wss?uid=swg21178966.

Table 26. IP information APARs for z/OS Communications Server

Title	Information APAR for V1R6
New Function Summary (both IP and SNA)	II13824
Quick Reference (both IP and SNA)	II13831
IP and SNA Codes	II13842
IP Sockets API Guide	II13844
IP Configuration Guide	II13826
IP Configuration Reference	II13827
IP Diagnosis	II13836
IP Messages Volume 1	II13838
IP Messages Volume 2	II13839
IP Messages Volume 3	II13840
IP Messages Volume 4	II13841
IPv6 Network and Application Design Guide	II13825
IP Programmer’s Guide and Reference	II13843
IP User’s Guide and Commands	II13832
IP System Admininstrator’s Commands	II13833

Information APARs for SNA documents

Table 27 lists information APARs for V1R6 SNA documents. For releases V1R7 and later, updates are available as technotes, which can be found at http://www.ibm.com/support/docview.wss?uid=swg21178966.

Table 27. SNA information APARs for z/OS Communications Server

Title	Information APAR for V1R6
New Function Summary (both IP and SNA)	II13824
Quick Reference (both IP and SNA)	II13831
IP and SNA Codes	II13842
SNA Customization	II13857
SNA Diagnosis, Vol. 1: Techniques and Procedures	II13852
SNA Diagnosis, Vol. 2: FFST Dumps and the VIT	II13853
SNA Messages	II13854
SNA Network Implementation Guide	II13849
SNA Operation	II13851
SNA Programming	II13858
SNA Resource Definition Reference	II13850
SNA Data Areas Volume 1	II13855
SNA Data Areas Volume 2	II13856

Other information APARs

Table 28 lists information APARs not related to documents.

Table 28. Non-document information APARs

Content	Number
Index to APARs that list recommended VTAM maintenance	II11220
Index to APARs that list trace and dump requests for VTAM problems	II13202
Index of Communication Server IP information APARs	II12028
Collecting TCPIP CTRACEs	II12014
CSM for VTAM	II13442
CSM for TCP/IP	II13951
DLUR/DLUS	II12986, II13456, and II13783
Documentation required for FTP server problems	II12925
Documentation required for OSA/2, OSA Express and OSA QDIO	II13016
DNS — common problems and solutions	II13453
Enterprise Extender	II12223
FTP client and FTP server TLS support	II13516
FTP problems	II12079
FTPing doc to z/OS Ssupport	II12030
Generic resources	II10986
HPR	II10953

Table 28. Non-document information APARs (continued)

Content	Number
iQDIO	II13142
LPR problems	II12022
MNPS	II10370
MPC and CTC	II01501
NCPROUTE problems	II12025
OMPROUTE	II12026
PASCAL API	II11814
Performance	II11710 II11711 II11712
Resolver	II13398 II13399 II13452
Socket API	II11996 II12020
SMTP problems	II12023
SNMP	II13477 II13478
SYSLOGD howto	II12021
TCPIP connection states	II12449
TN3270E Telnet server	II11574 II13135
TN3270E Telnet server SSL common problems	II13369

Appendix H. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`www.ibm.com/servers/eserver/zseries/zos/bkserv/`

Notices

IBM may not offer all of the products, services, or features discussed in this document. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California. Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Portions of this publication relating to RPC are Copyright © Sun Microsystems, Inc., 1988, 1989.

Some portions of this publication relating to X Window System** are Copyright © 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, and the Massachusetts Institute Of Technology, Cambridge, Massachusetts. All Rights Reserved.

Some portions of this publication relating to X Window System are Copyright © 1986, 1987, 1988 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute the M.I.T., Digital Equipment Corporation, and Hewlett-Packard Corporation portions of this software and its documentation for any purpose without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the names of M.I.T., Digital, and Hewlett-Packard not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T., Digital, and Hewlett-Packard make no representation about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

Copyright © 1983, 1995-1997 Eric P. Allman

Copyright © 1988, 1993 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement: ″This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)″. The word ’cryptographic’ can be left out if the routines from the library being used are not cryptographic related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include acknowledgement:

   ″This product includes software written by Tim Hudson (tjh@cryptsoft.com)″

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License.]

This product includes cryptographic software written by Eric Young.

Copyright © 1999, 2000 Internet Software Consortium.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED ″AS IS″ AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

X Window System is a trademark of The Open Group.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

You can obtain softcopy from the z/OS Collection (SK3T-4269), which contains BookManager and PDF formats.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking
AFS
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
AS/400
BookManager
C/370
CICS
CICS/ESA
C Set ++
Common User Access
CUA
DB2
DFSMS
DFSMSdfp
DFSMShsm
DPI
ESCON
eServer
ES/9000
FFST
FICON
First Failure Support Technology
GDDM
IBM
ibm.com
IBMLink
IMS
IMS/ESA
HiperSockets
Language Environment
Micro Channel
Multiprise
MVS
MVS/DFP
MVS/ESA
MVS/SP
NetView
Network Station
Nways
OfficeVision
OS/2
OS/390
Parallel Sysplex
PROFS
pSeries
RACF
Redbooks
RETAIN
REXX
RISC System/6000
RMF
RS/6000
S/370
S/390
S/390 Parallel Enterprise Server
SAA
SecureWay
SET
SiteCheck
SP
System/360
System/370
System/390
System z
System z9
Tivoli
Tivoli Enterprise Console
VM/ESA
VSE/ESA
VTAM
WebSphere
z9
z/Architecture
z/OS
z/VM
zSeries
400

The following terms are trademarks of other companies:

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

PostScript is a registered trademark of Adobe Systems Incorporated in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Bibliography

z/OS Communications Server information

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

z/OS Communications Server documentation is available:

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

z/OS Communications Server library

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

Updates to documents are available on RETAIN® and in information APARs (info APARs). See Appendix G, "Information APARs and technotes," on page 907 for a list of the documents and the info APARs associated with them.

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

Planning

Title	Number	Description
z/OS Communications Server: New Function Summary	GC31-8771	This document is intended to help you plan for new IP for SNA function, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC31-8885	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC31-8775	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Reference*.

Title	Number	Description
z/OS Communications Server: IP Configuration Reference	SC31-8776	This document presents information for people who want to administer and maintain IP. Use this document in conjunction with the *z/OS Communications Server: IP Configuration Guide*. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • SMF records • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC31-8777	This document presents the major concepts involved in implementing an SNA network. Use this document in conjunction with the *z/OS Communications Server: SNA Resource Definition Reference*.
z/OS Communications Server: SNA Resource Definition Reference	SC31-8778	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document in conjunction with the *z/OS Communications Server: SNA Network Implementation Guide*.
z/OS Communications Server: SNA Resource Definition Samples	SC31-8836	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: IP Network Print Facility	SC31-8833	This document is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC31-8780	This document describes how to use TCP/IP applications. It contains requests that allow a user to log on to a remote host using Telnet, transfer data sets using FTP, send and receive electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC31-8781	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC31-8779	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SX75-0124	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
z/OS Communications Server: SNA Customization	SC31-6854	This document enables you to customize SNA, and includes the following: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs

Title	Number	Description
z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference	SC31-8788	This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC31-8807	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS® using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC31-8830	This document is for programmers who want application programs that use the IMS™ TCP/IP application development services provided by IBM's TCP/IP Services.
z/OS Communications Server: IP Programmer's Guide and Reference	SC31-8787	This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC31-8829	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC31-8811	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC31-8810	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.
z/OS Communications Server: CSM Guide	SC31-8808	This document describes how applications use the communications storage manager.

Title	Number	Description
z/OS Communications Server: CMIP Services and Topology Agent Guide	SC31-8828	This document describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The document provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis

Title	Number	Description
z/OS Communications Server: IP Diagnosis Guide	GC31-8782	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and *z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT*	GC31-6850 GC31-6851	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
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	SC31-8790	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)	SC31-8783	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)	SC31-8784	This volume contains TCP/IP messages beginning with EZB or EZD.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC31-8785	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)	SC31-8786	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC31-8791	This document describes codes and other information that appear in z/OS Communications Server messages.

Index

Communicating Your Comments to IBM

If you especially like or dislike anything about this document, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

Please send your comments to us in either of the following ways:

- If you prefer to send comments by FAX, use this number: 1+919-254-1258
- If you prefer to send comments electronically, use this address:
  - comsvrcf@us.ibm.com
- If you prefer to send comments by post, use this address:

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

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.

Program Number: 5694–A01

Printed in USA

SC31-8781-07

Spine information:

IBM

z/OS Communications Server

z/OS V1R9.0 Comm Svr: IP Sys Admin Commands

Version 1
Release 9