VSI OpenVMS

VSI TCP/IP Administrator's Reference

Document Number: DO-DVTADR-01B

Publication Date: January 2020

This document provides information to configure and manage VSI TCP/IP for the experienced system manager.

Revision Update Information: This guide supercedes the VSI TCP/IP Administrator's Reference, Version 10.5.

Operating System and Version: VSI OpenVMS Version 8.4-2L1 or higher

Software Version: VSI TCP/IP for OpenVMS Version 10.6

VMS Software, Inc., (VSI)
Bolton, Massachusetts, USA

VSI TCP/IP Administrator's Reference:

Legal Notice

The VSI OpenVMS documentation set is available on CD.

Preface

1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

VSI seeks to continue the legendary development prowess and customer-first priorities that are so closely associated with the OpenVMS operating system and its original author, Digital Equipment Corporation.

2. Intended Audience

This manual is intended for any experienced system manager who will be using VSI TCP/IP. It describes how to configure and manage VSI TCP/IP and contains information about using commands for the following:

- DCL prompt
- DECNET-CONFIG command line
- MAIL-CONFIG command line
- NET-CONFIG command line
- NFS-CONFIG command line
- Network Terminal Device Control Program (NTYCP)
- PRINTER-CONFIG command line
- SERVER-CONFIG command line

3. Typographical Conventions

The following conventions are used in this manual:

Convention	Meaning
Ctrl/*X*	A sequence such as **Ctrl/*x*** indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *X*	A sequence such as **PF1 *X*** indicates that you must first press and release the key labeled PF1 and then press and release another key (***x***) or a pointing device button.
Enter	In examples, a key name in bold indicates that you press that key.
...	A horizontal ellipsis in examples indicates one of the following possibilities:- Additional optional arguments in a statement have been omitted.- The preceding item or items can be repeated one or more times.- Additional parameters, values, or other information can be entered.

Convention	Meaning
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one. In installation or upgrade examples, parentheses indicate the possible answers to a prompt, such as: `Is this correct? (Y/N) [Y]`
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for directory specifications and for a substring specification in an assignment statement. In installation or upgrade examples, brackets indicate the default answer to a prompt if you press **Enter** without entering a value, as in: `Is this correct? (Y/N) [Y]`
|	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold type	Bold type represents the name of an argument, an attribute, or a reason. In command and script examples, bold indicates user input. Bold type also represents the introduction of a new term.
italic type	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (**/PRODUCER=*name***), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Example`	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies website addresses, UNIX command and pathnames, PC-based commands and folders, and certain elements of the C programming language.
-	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes-binary, octal, or hexadecimal-are explicitly indicated.

4. VSI TCP/IP Support

VSI supports VSI TCP/IP running on VSI OpenVMS Integrity Version 8.4-2L1 (or higher) only. Please contact your support channel for help with this product. Users who have OpenVMS support contracts through VSI can contact support@vmssoftware.com [mailto:support@vmssoftware.com] for help with this product. Users who have OpenVMS support contracts through HPE should contact their HPE Support channel for assistance.

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: `<docinfo@vmssoftware.com>`.

6. How to Order Additional Documentation

For information about how to order additional documentation, email the VSI OpenVMS information account: `<info@vmssoftware.com>`. We will be posting links to documentation on our corporate website soon.

Chapter 1. DCL Command Reference

This chapter describes administration commands that you can run from the DCL prompt.

Command Summary

Describes administrative commands of VSI TCP/IP that are available at the DCL prompt.

Table 1.1. VSI TCP/IP Command Summary

Command	Description
IP ACCOUNTING	Processes the accounting file that FTP and SMTP can write.
IP CHECK	Tests the VSI TCP/IP configuration.
IP CONFIGURE **/CONFIGURATION_FILE** **/DECNET** **/MAIL** **/NETWORK** **/PRINTERS** **/SERVERS** **/SERVER_IMAGE**	Specifies the configuration file read by the DECNET-, PRINTER-, or SERVER-CONFIG utility. Invokes the DECnet Circuit Configuration Utility (DECNET-CONFIG). Invokes the Electronic Mail Configuration Utility (MAIL-CONFIG). Invokes the Network Interface Configuration Utility (NET-CONFIG). Invokes the Printer Configuration Utility (PRINTER-CONFIG). Invokes the Server Configuration Utility (SERVER-CONFIG). Specifies the master server image associated with the server configuration file.
IP DIG	Tests the domain name service (DNS) system.
IP DNSKEYGEN	Generates and maintains keys for DNS Security (DNSSEC) within the DNS.
IP DNSSIGNER	Signs zone files for DNS Security (DNSSEC) within the DNS (Domain Name System).
IP FONT COMPILE	Compiles an ASCII BDF (bitmap distribution format) font file into a binary PCF (portable compiled format) file.
IP FONT INFO	Displays font server information.
IP FONT LIST	Lists font names and font information.
IP FONT MKFONTDIR	Creates a `DECW$FONT_DIRECTORY.DAT` file when adding fonts.
IP FONT SHOW	Displays font data.
IP FONT UNCOMPILE	Uncompiles a PCF file into an ASCII BDF file.
IP GATED /CHECK	Checks the syntax of a GateD configuration file.
IP GATED /DUMP	Tells GateD to dump internal state into a text file.
IP GATED /LOAD	Loads new configuration file.

Command	Description
IP GATED /SET /TRACE	Controls tracing in GateD.
IP GATED /SHOW /OSPF	Queries OSPF routers.
IP GATED /SHOW /RIP	Request all routes known by a RIP gateway.
IP GATED /SHOW /TRACE	Queries tracing in GateD.
IP GATED /STOP	Tells the GateD process to halt in an orderly manner.
IP GATED /TOGGLE_TRACING	Toggles GateD tracing on and off.
IP GATED /UPDATE_INTERFACES	Tells the GateD process to rescan the network interfaces.
IP HOST_TABLE GET	Retrieves a `HOSTS.TXT` file.
IP HOST_TABLE INSTALL	Installs host tables as global sections.
IP IPP SHOW	Allows a user to learn the capabilities supported by an IPP server.
IP LOAD	Loads and invokes the network image.
IP NETCONTROL	Sends commands to IP$SERVER internal services.
IP NFSDISMOUNT	Dismounts a locally-mounted remote NFS file system.
IP NFSMOUNT	Mounts a remote NFS file system so it can be used locally.
IP NSLOOKUP	Sends a test query to DNS.
IP NSUPDATE	Performs dynamic updates to the domain name service (DNS) server.
IP PING	Tests connections by sending ICMP echo requests.
IP PING6	Tests connections by sending ICMPv6 echo requests.
IP RMTALLOC	Allocates a remote tape drive or CD-ROM for access by a single process.
IP RWALL	Sends a message to all system users.
IP SET /ARP	Changes ARP tables.
IP SET /DECNET	Configures DECnet devices to run DECnet-over-UDP circuits.
IP SET /INTERFACE	Sets parameters for network devices
IP SET /ROUTE	Specifies static IP routing.
IP SET /TIMEZONE	Specifies the local timezone name.
IP TCPDUMP	Decodes network packets selected by a boolean expression.
IP TCPVIEW	Traces packets and interprets the results.
IP TRACEROUTE	Determines the route to the specified host.
IP TRACEROUTE /IPv6	Determines the route to the specified host for IPv6.
IP X11DEBUG	Performs tests on the most common causes of problems when running X11 clients over VSI TCP/IP.

IP ACCOUNTING

IP ACCOUNTING — Processes the accounting file that session accounting writes for SMTP and for FTP. It extracts the selected records from it and either displays it on the user's terminal or sends it to the specified output file.

Format

`$ IP ACCOUNTING /INPUT=`*`filename`*` /SINCE=`*`first_date_to_include`*

Qualifiers

`/BEFORE=`*`latest_date_to_include`*

`/CSV`

Makes the output file a Comma Separated Values file that can be imported into an Excel-type document for processing.

`/INPUT=`*`accounting_file_name`*

`/OUTPUT=`*`output_file_name`*

`/PROTOCOL=`*`(MAIL,SMTP,FTP)`*

These are the protocols to include.

`/SINCE=`*`first_date_to_include`*

IP CHECK

IP CHECK — Invokes the VSI TCP/IP configuration test utility to perform one or more checks for common VSI TCP/IP configuration problems. Requires CMKRNL, SYSPRV, and WORLD privileges.

Format

`IP CHECK `*`[test,...]`*

Parameter

`test`

Specifies the name of a test to be performed. Valid test names are ARP, BROADCASTS, DATABASES, HOST_NAME, HOST_TABLE, INTERFACES, LICENSE, MISCELLANEOUS, PARAMETERS, PROTOCOL_ERRORS, ROOT_NAMESERVERS, ROUTES, and VERSION. You can specify multiple tests by separating the names with commas. If you do not specify a test parameter, all tests are performed.

The host name check verifies that the address associated with the local host name matches one of the interface addresses.

Qualifiers

`/IGNORE_ERRORS`

`/NOIGNORE_ERRORS (default)`

IP CHECK usually stops when it encounters an error. Specify this qualifier to force **IP CHECK** to continue testing even after an error is encountered.

`/OUTPUT=`*`file-spec`*

`/NOOUTPUT`

IP CHECK usually displays all output on the standard error output device. Specify this qualifier to either redirect output to the specified file or turn output off altogether.

`/VERBOSE`

`/NOVERBOSE (default)`

Causes **IP CHECK** to display more information about the tests it performs. By default, it only displays a message when it encounters an error or if all tests pass.

IP CONFIGURE

IP CONFIGURE — Invokes one of the VSI TCP/IP configuration utilities which are interactive programs that maintain network configuration information. If you do not specify a configuration utility with a qualifier, the network interface configuration utility (NET-CONFIG) is invoked.

Format

`IP CONFIGURE` *`[/qualifier(s)]`*

Qualifiers

`/CONFIGURATION_FILE=`*`config_file`*

Used with the **/DECNET**, **/PRINTERS**, or **/SERVERS** qualifier, specifies the configuration file read by the corresponding utility.

`/DECNET`

Invokes the DECnet Configuration Utility (DECNET-CONFIG) that lets you view and alter the configuration of DECnet-over-IP services. If used with the **/CONFIGURATION_FILE** qualifier, DECNET-CONFIG reads the specified configuration file (by default, `IP$:DECNET-CIRCUITS.COM`).

`/INTERFACES`

Invokes the Network Interface Configuration Utility (NET-CONFIG) that lets you view and alter the configuration of network interfaces, routing, and host name lookup. If used with the **/CONFIGURATION_FILE** qualifier, NET-CONFIG reads the specified configuration file (by default, `IP$:NETWORK_DEVICES.CONFIGURATION`).

`/MAIL`

Invokes the Electronic Mail Configuration Utility (MAIL-CONFIG) that lets you view and alter SMTP configuration. If used with the **/CONFIGURATION_FILE** qualifier, MAIL-CONFIG reads the specified configuration file (by default, `IP$COMMON_ROOT:[IP]START_SMTP.COM`).

`/NETWORK`

Invokes the Network Interface Configuration Utility (NET-CONFIG) that lets you view and alter the configuration of network interfaces, routing, and host name lookup. If used with the **/CONFIGURATION_FILE** qualifier, NET-CONFIG reads the specified configuration file (by default, `IP$:NETWORK_DEVICES.CONFIGURATION`).

IP CONFGURE /NETWORK now has the command:

SET SNMP-AGENTX TRUE to enable SNMP Agent X service.

SET SNMP-AGENTX FALSE to disable SNMP Agent X service.

A line displays in the output of the **SHOW** command if SNMP Agent X subagents are enabled.

`/NFS`

Invokes the VSI TCP/IP NFS Server Configuration Utility (NFS-CONFIG) for the VSI TCP/IP NFS Server option.

If used with the **/CONFIGURATION_FILE** qualifier, NFS-CONFIG reads the specified configuration files: `NFS_EXPORT.DAT, NFS_GROUP.DAT, NFS_MNTLST.DAT,` and `NFS_PROXY.DAT`.

`/NOT`

Invokes the NOT Configuration Utility NOT-CONFIG for DECnet applications services (formerly known as Phase/IP). DECnet application services allow you to run applications designed to use DECnet using TCP/IP instead. DECnet application services provide the DECnet API (Application Programming Interface) across TCP seamlessly, without DECnet protocols or software, and without the additional overhead of running both protocol stacks.

`/PRINTERS`

Invokes the Printer Configuration Utility (PRINTER-CONFIG) of VSI TCP/IP that lets you view and alter the configuration of VSI-based print services that are based on VSI TCP/IP. If used with the **/CONFIGURATION_FILE** qualifier, PRINTER-CONFIG reads the specified configuration file (by default, `IP$:REMOTE-PRINTER-QUEUES.COM`).

`/SERVERS`

Invokes the Service Configuration Utility (SERVER-CONFIG) of VSI TCP/IP that lets you view and alter the configuration of VSI TCP/IP services. If used with the **/CONFIGURATION_FILE** qualifier, SERVER-CONFIG reads the specified configuration file (by default, `IP$:SERVICES.MASTER_SERVER`).

`/SERVER_IMAGE=server_image_file`

Used with the /SERVERS qualifier, server_image_file specifies the VSI TCP/IP master server image associated with the server configuration file. This file is used by SERVER-CONFIG to determine which network services are available. If not specified, SERVER-CONFIG uses `IP$:SERVER.EXE`.

IP DIG

IP DIG — Similar to NSLOOKUP, DIG tests the domain name service (DNS) system. It uses the DNS resolver to send queries to the DNS server and prints out the response. DIG executes a single command or reads commands from a file (in "batch mode").

Format

`IP DIG` [*name* [type[*class*]]]

Description

DIG can be used with the UNIX-style syntax by defining it as a foreign command:

```
$ DIG :== $IP$:DIG.EXE
```

Both the UNIX-style options and the OpenVMS qualifiers are listed below.

Parameters

[*name*]

Specifies a host or domain name.

Note

You must specify fully-qualified names. DIG will not append any domain names.

[*type*]

Specifies which **TYPE** resource records are asked for. The default is A (address records). Valid values are the same as for the **NSLOOKUP /TYPE** qualifier (see Table 1.33).

[*class*]

Specifies which CLASS resource records are asked for. The default is IN (internet records). Valid values are `ANY`, `IN`, `CHAOS`, and `HESIOD`.

Qualifiers

`+[no]addit`

`/ADDITIONAL (default)`

`/NOADDITIONAL`

Tells the resolver to print the additional section of the reply.

`-x` *`ip-address`*

`/ADDRESS=`*`ip-address`*

Convenient form to specify an inverse address mapping query. For example, **IP DIG / ADDRESS=10.5.64.1** is equivalent to **IP DIG 1.64.5.10.IN-ADDR.ARPA ANY**.

`+[no]answer`

`/ANSWER (default)`

`/NOANSWER`

Tells the resolver to print the answer section of the reply.

`+[no]author`

`/AUTHORITY (default)`

`/NOAUTHORITY`

Tells the resolver to print the authority section of the reply.

`-c` *`recordclass`*

`/CLASS=`*`recordclass`*

Specifies which CLASS resource records are asked for. Alternative to specifying the class parameter. The *`recordclass`* value may be either the integer value of the class or the name of the class (ANY, IN, CHAOS, HESIOD). The default is IN (internet records).

`+[no]cmd`

`/CMD (default)`

`/NOCMD`

Tells DIG to echo parsed arguments from the command.

`+[no]debug`

`/DEBUG`

`/NODEBUG (default)`

Causes the resolver to print debugging information.

`+[no]d2`

`/DEBUG2`

`/NODEBUG2 (default)`

Causes the resolver to print additional, less useful debugging information.

`-envsav`

`/ENVSAVE`

Specifies that the DIG environment (defaults, print options, etc.), after all of the arguments are parsed, should be saved to a file to become the default environment. This is useful if you do not like the standard set of defaults and do not desire to include a large number of options each time DIG is used. The environment consists of resolver state variable flags, timeout, and retries as well as the flags detailing DIG output. If the logical name `LOCALDEF` is set to the name of a file, this is where the default DIG environment is saved. If not, the file `DIG.ENV` is created in the current default directory.

Each time DIG is executed, it looks for `DIG.ENV` or the file specified by `LOCALDEF`. If such a file exists, then the environment is restored from this file before any arguments are parsed.

`-envset`

`/ENVSET`

This qualifier only affects batch query runs. When -envset is specified on a line in a DIG batch file, the DIG environment after the arguments are parsed becomes the default environment for the duration of the batch file, or until the next line which specifies -envset. Remember that commands in the DIG batch file must be in UNIX-style syntax.

`-f filename`

`/FILE=filename`

Causes DIG to run in batch mode, executing the commands in the specified file. The commands in this file must be in the UNIX-style syntax.

`+[no]Header`

`/HEADER (default)`

`/NOHEADER`

Tells the resolver to print basic header information.

`+[no]header`

`/HFLAGS (default)`

`/NOHFLAGS`

Tells the resolver to print header flags.

`+[no]ignore`

`/IGNORE`

`/NOIGNORE (default)`

Tells the resolver to ignore truncation in responses.

`+[no]ko`

`/KEEPOPEN`

`/NOKEEPOPEN (default)`

If using virtual circuits (TCP), keeps the connection open.

`-k` *`keydir+keyname`*

`/KEY=(KEYNAME=`*`key`* [,KEYDIR=*`directory`*]

Specifies a TSIG key for DIG to use to sign its queries. The default value for KEYDIR is the current default directory.

Note

On UNIX systems, the syntax is *`keydir:keyname`*. On OpenVMS, the colon is replaced by a plus sign (+). The *`keyname`* must be specified to match the key and private filenames, with periods instead of dollar signs. This may not match the domain name if DNSKEYGEN had to abbreviate it to fit into an OpenVMS file name.

`+pfand=`*`number`*

`/PFAND=`*`number`*

Causes DIG to do a bitwise-AND of the print flags with the specified value.

`+pfdef`

`/PFDEF (default)`

`/NOPFDEF`

Sets the print flags to the default.

`+pfmin`

`/PFMIN`

`/NOPFMIN (default)`

Sets the print flags to the minimum.

`+pfor=`*`number`*

`/PFOR=`*`number`*

Causes DIG to do a bitwise-OR of the print flags with the specified value.

`+pfset=`*`number`*

`/PFSET=`*`number`*

Sets the print flags to the specified value.

`-P `*`ping-command`*

`/PING[=`*`ping-command`*`]`

Causes DIG to execute a ping command to the queried nameserver after the query returns, for response time comparison. If the optional *`ping-command`* is present, it is used as the ping command. The default ping command is "IP PING".

`-p `*`port`*

`/PORT=`*`port`*

Specifies a port other than the standard nameserver port of 53.

`+[no]qr`

`/QUERY`

`/NOQUERY (default)`

Tells the resolver to print the outgoing query.

`+[no]ques`

`/QUESTION (default)`

`/NOQUESTION`

Tells the resolver to print the question section of the reply.

`+[no]recurse`

`/RECURSE (default)`

`/NORECURSE`

Requests that the name server use recursion to answer the query.

`+[no]reply`

`/REPLY (default)`

`/NOREPLY`

Tells the resolver to print the reply.

`+retry=`*`retrycount`*

`/RETRY=`*`retrycount`*

Specifies the number of retries the resolver makes when querying a name server via UDP. The default is 4.

`@`*`server`*

`/SERVER=`*`server`*

Specifies the nameserver to query. May be specified as either a domain name or a dot-notation internet address. If a domain name is specified, DIG looks up the name using the default nameserver. If /**SERVER** is not specified, the default is to use the system's default nameserver.

`+[no]stats`

`/STATS (default)`

`/NOSTATS`

Tells the resolver to print query statistics.

`-[no]stick`

`/STICKY`

`/NOSTICKY (default)`

This qualifier only affects batch query runs. -stick specifies that the DIG environment (as read initially or set by -envset switch) is to be restored before each query (line) in a DIG batch file. The default -

nostick means that the DIG environment does not stick, hence options specified on a single line in a DIG batch file will remain in effect for subsequent lines (i.e., they are not restored to the "sticky" default). Remember that commands in the DIG batch file must be in UNIX-style syntax.

+time=*seconds*

/TIMEOUT=*seconds*

Specifies a different period to wait for responses. The default is 4 seconds.

-T *seconds*

/TIMEWAIT=*seconds*

Causes DIG to wait the specified number of seconds between the start of successive queries when running in batch mode. Can be used to keep two or more batch DIG commands running roughly in sync. The default is 0.

-t *recordtype*

/TYPE=*recordtype*

Specifies which **TYPE** resource records are asked for. Alternative to specifying the type parameter. The *recordtype* value may be either the integer value of the type or the name of the type (see Table 1.33). The default is A (address records).

+[no]vc

/VC

/NOVC (default)

Specifies that the resolver use virtual circuits (TCP) instead of datagram (UDP) queries.

Example

The following is an example of the default DIG output:

```
$ IP dig www.peh.com

; <<>> DiG 8.3 <<>> WWW.PEH.COM
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 2
;; QUERY SECTION:
;;             WWW.PEH.COM, type = A, class = IN

;; ANSWER SECTION:
WWW.PEH.COM.   2H IN CNAME   peh.com.
peh.com.       2H IN A       209.196.131.83

;; AUTHORITY SECTION:
peh.com.       2H IN NS      ns1.pbi.net.
peh.com.       2H IN NS      ns2.pbi.net.

;; ADDITIONAL SECTION:
```

```
ns1.pbi.net.   2D IN A       206.13.28.11
ns2.pbi.net.   2D IN A       206.13.29.11

;; Total query time: 14289 msec
;; FROM: bite.process.com to SERVER: default -- 127.0.0.1
;; WHEN: Thu Jun  1 14:52:49 2002
;; MSG SIZE  sent: 29  rcvd: 141
```

IP DNSKEYGEN

IP DNSKEYGEN — (DNS Key Generator) is a tool to generate and maintain keys for DNS Security (DNSSEC) within the DNS (Domain Name System).

Format

`IP DNSKEYGEN` *`name`*

Description

DNSKEYGEN can generate public and private keys to authenticate zone data, and shared secret keys to be used for Request/Transaction Signatures. **DNSKEYGEN** can be used with the UNIX-style syntax by defining it as a foreign command:

```
$ DNSKEYGEN :== $IP$:DNSSEC-KEYGEN.EXE;
```

Both the UNIX-style options and the OpenVMS qualifiers are listed below.

DNSKEYGEN stores each key in two files: `K`*`name.alg-footprint`*`-private` and `K`*`name.alg-footprint`*`-key`. *`name`* is the domainname with the periods replaced by dollar signs. The first file contains the private key in a portable format. The second file contains the publickey in the DNS zone file format:

`name` `IN KEY` *`flags protocol algorithm exponent`*`|`*`module`*

If the domain name is too long for an OpenVMS filename, it is truncated to fit and the last six characters are replaced by unique digits. The full domain name can be found inside the key file.

Parameters

`-n name`

`name`

Specifies the domain name to generate the key for.

Qualifiers

`-D` *`size`*

`/DSA_DSS=`*`size`*

`-H` *`size`*

`/HMAC_MD5=`*`size`*

`-R` *`size`*

`/RSA=`*`size`*

These flags specify the type of key to generate. **You must specify one and only one of these**.

If **/DSA_DSS** is specified, **DNSKEYGEN** generates a DSA/DSS key. *`size`* must be one of: 512, 576, 640, 704, 768, 832, 896, 960, or 1024.

If **/HMAC_MD5** is specified, **DNSKEYGEN** generates an HMAC-MD5 key. *`size`* must be between 128 and 504.

If **/RSA** is specified, **DNSKEYGEN** generates an RSA key. *`size`* must be between 512 and 4096.

`-F`

`/LARGE_EXPONENT`

Used for RSA only. If specified, **DNSKEYGEN** uses a large exponent for key generation.

`-z`

`/ZONE_KEY`

`-h`

`/HOST_KEY`

`-u`

`/USER_KEY`

These flags define the type of key being generated. **You must specify one and only one of these.**

- Zone (DNS validation) key
- Host (host or service) key
- User (e.g., email) key

`-a`

`/NOAUTHENTICATION`

Indicates that the key CANNOT be used for authentication.

`-c`

`/NOENCRYPTION`

Indicates that the key CANNOT be used for encryption.

`-p` *`num`*

`/PROTOCOL=`*`num`*

Sets the key's protocol field to *`num`*. If **/ZONE_KEY (-z)** or /**HOST_KEY (-h)** is specified, the default is 3 (DNSSEC); otherwise, the default is 2 (EMAIL). Other accepted values are 1 (TLS), 4 (IPSEC), and 255 (ANY).

-s *num*

/STRENGTH=*num*

Sets the key's strength field to *num*; the default is 0.

Example

The following example generates an RSA key.

```
$ IP DNSKEYGEN/RSA=512/ZONE_KEY zone.example
** Adding dot to the name to make it fully qualified domain name**
Generating 512 bit RSA Key for ZONE.EXAMPLE.

Generated 512 bit Key for ZONE.EXAMPLE. id=49663 alg=1 flags=257
```

DNSKEYGEN generates the following (for example):

File `KZONE$EXAMPLE$.001-49663-KEY`:

```
ZONE.EXAMPLE. IN KEY 257 3 1 AQOojr81q9PfmQXCUAJOoMu3CYaS78RZnhiV/
uAfSbzZusWYLSeVF47OwZlmgwclswZoaM5NSuzFX3w5RDIEwf9c
```

File `KZONE$EXAMPLE$.001-49663-PRIVATE`:

```
Private-key-format: v1.2
Algorithm: 1 (RSA)
Modulus: qI6/NbPT35kGwlACTqDLtwmGku/
EWZ4Ylf7gH0m82arFmC0nlReOjsGJZoMHJbMGaGjOTUrsxV98OUQyAMH/Ww==
PublicExponent: Aw==
PrivateExponent: cF8qI8036mZD1uABjcCHz1uvDJ/
YO767Dqmqv4Z95ntuhY7uIMmn8zy0Ur9kj/7P5Dvpu7ZG91ZtuQ1YhWAMyw==
Prime1: 2IQQP2+DvU/G0038OCoji00NDQHA0az8lDV1fh8Qf9k=
Prime2: x0vGgXRlWVIfp5xnuCORP0UB4rK3sKVhQ246rx2hbFM=
Exponent1: kFgK1PQCfjUvN4lS0BwXtN6Is1aBNnNTDXj4/r9gVTs=
Exponent2: hN0vABhDjja/xRLv0Be2Kl4BQcv6dcOWLPQnH2kWSDc=
Coefficient: YQGEh81Y720mRfAV/tEs3eWKd11Mm10b5R4lFjVwtAU=
```

IP DNSSIGNER

IP DNSSIGNER — **DNSSIGNER** is a tool to sign zone files for DNS Security (DNSSEC) within the DNS (Domain Name System). **DNSSIGNER's** job is to read the data of one zone of DNS data, and perform the necessary work to produce the data for a secured zone.

Format

```
IP DNSSIGNER
```

Description

DNSSIGNER can be used with the UNIX-style syntax by defining it as a foreign command:

```
$ DNSSIGNER :== $IP$:DNSSEC-SIGNZONE.EXE;
```

Both the UNIX-style options and the OpenVMS qualifiers are listed below.

You can get help on the UNIX-style options using:

- $ dnssigner -h: ! for short help
- $ dnssigner -help: ! for long help

Signing is done on a zone-by-zone basis, regardless of the relationship of zones to name servers. **DNSSIGNER** is designed to operate in a dynamic environment, including those in which secret keys are not available to all of those covering a zone, and where information may be arriving after the beginning of the signing process. **DNSSIGNER** makes an effort to retain valid signatures instead of computing new signatures.

Using traditional BIND DNS zone master files, there are two things necessary as input to use **DNSSIGNER** to sign a zone. One is the names of the input files and the other is the names of the keys to use. There are two kinds of data files used as input to the signing process. The standard zone master file, and a master file introduced by DNSSEC called the parent file. A parent file contains output from the signing of the parent zone, most importantly the signature by the parent of the zone's keys.

Input/output details

The default input zone is START-ZONE. A different zone input file can be specified with /**ZONE=(INPUT=*`filename`***) (-zi). There is no default input parent file. A parent file can be specified with **/PARENT=(INPUT=*`filename`*) (-pi)**.

The default output files are `FINISH-ZONE` and `FINISH-PARENT`. /**ZONE=(OUTPUT=*`filename`***) (-zo) changes the name of the zone output file, and /**PARENT=(OUTPUT=*`filename`*) (-po)** changes the name of the parent file generated by the zone.

Parent file handling

There are two forms of parent file generation. One form is to place all of the parent files in one file (good for zones with many delegations), the other is to make a separate file for each delegation. Since it is easier to erase one file than potentially thousands, **DNSSIGNER** defaults to the single signer file.

/PARENT=NOBULK (-no-p1) turns single parent file generation off, **/PARENT=BULK (-p1)** turns it on. As mentioned earlier **/PARENT=OUTPUT=*`filename`* (-po)** sets the name of the single parent file (default `FINISH-PARENT`.).

/PARENT=INDIVIDUAL (-ps) turns on individual parent files, **/PARENT=NOINDIVIDUAL (-no-ps)** turns it off. **/PARENT=DIRECTORY=*`spec`* (-pd)** sets the directory into which the individual files are put (default is the current working directory).

NXT details

/**NONXT (-no-n)** turns off RFC 2065 NXT processing. /**NXT (-n)** (default) turns on RFC 2065 NXT processing.

Key details

Use the -k1 flag (the **/SIG=(KEY=( ))** qualifier) or the -ks flag (no OpenVMS-style equivalent) to specify a key. -k1 is followed by a domain name owner of a key, the algorithm, and the key id. -ks is followed by a sequence of names, algorithms, and key ids until the end of the command line.

SIG expiration details

There are two time durations that are important to the handling of signatures. One is the duration until a newly generated signature is set to expire. The other is the duration in which existing signatures will be considered to be expired.

/SIG=DURATION=*`ttl`* (-dur) sets the duration for which a signature is valid.

The time included in the SIG RR expiration field is the current absolute time plus the duration. Wrapping around 32 bits is not a problem, as time is considered to be "circular."

/SIG=PURGE_PERIOD=*`ttl`* (-pt) sets the period into the future in which SIGs expiring then are considered to have expired. Any signature that has an expiry time in the past of the current time is thrown out, as well as signature whose expiry time falls into the span between now and the purge period duration. The past is considered to be the time from now back to 2 to the 31st seconds ago; the rest is the future.

Qualifiers

This section describes the syntax of all flags. The meanings can be found in RFC 2065 and the drafts associated with the **DNSSEC** working group.

`-[no-]bind`

`/[NO]BIND`

/BIND (-bind) instructs **DNSSIGNER** to use BIND's extended TTLs and KEY flags when writing files. This is the default. Use **/NOBIND (-no-bind)** to turn this feature off. In this case TTLs and flags are written as numeric values.

`-l option`

`/DEBUG=option`

Specifies the level of output (debug) messages that **DNSSIGNER** should print. Specify one of the following levels: (UNIX-syntax equivalents are also shown)

-l 7 -l deb DEBUGGER	Print source code locations, errors, and warnings.
-l 10 -l dev DEVELOPER	Print source code locations and cryptography messages.
-l 1 -l m MINIMAL	Print just errors.
-l 4 -l u USER	Print errors and warnings. This is the default.

`-[no-]n`

`/[NO]NXT`

/NXT (-n) (default) instructs **DNSSIGNER** to generate NXT RRs for the zone, signing them with the keys that sign the SOA record. (If none sign the SOA, no NXT's are signed.). Use /NONXT (-no-n) to turn this feature off.

`-or` *`domain`*

`/ORIGIN=`*`domain`*

This is equivalent to the **$ORIGIN** ***`domain`*** directive in the zone file, except that the terminating period is not needed in the domain name. Specifying an origin is only mandatory for the root zones and other zones using relative names in the zone files. It is recommended that the **$ORIGIN** ***`domain`*** directive be put in the data file. By default, this is unspecified.

`/PARENT=(`*`keyword`*`[,...])`

Specifies options related to parent zone files. Possible keywords (and their UNIX-syntax equivalents) are as follows.

There are two ways in which parent files are made: individual and bulk. The two methods use independent keywords. Both can be used, neither can be used, or just one. By default, the bulk approach is used.

-[no-]p1 [NO]BULK	BULK (-p1) (default) tells **DNSSIGNER** to place all of the generated parent data for the zone's delegation points into one file. Separating lines are added to identify the start and end of the information destined for individual zones. Use NOBULK (-no-p1) to turn this feature off.
-pd *`directory`* DIRECTORY=*`directory`*	Specifies the directory to put individual parent files into. The default is the current default directory.
-[no-]ps [NO]INDIVIDUAL	INDIVIDUAL (-ps) tells DNSSIGNER to place the generated parent data into individual files, named *zone*.PARENT. For large delegated zones, there will be many files. The default is NOINDIVIDUAL (-no-ps).
-pi *`file`*INPUT=*`file`*	Specifies the parent file received from the parent zone to be used as input to this zone. If specified, all records that would conflict with it (apex upper NXT, KEYs, and SIGs for these) are dropped. If the UP policy is specified, then the parent's KEY, NS, and glue are also dropped. The default is to have no parent file.
-pa *`domain`* NAME=*`domain`*	Specifies the apex's parent zone. If the keys for this zone are known and the UP policy is used, the apex zone keys sign the key. If UP is used and this is not specified, then DNSSIGNER acts as if it does not otherwise know the parent's identity. This is equivalent to the $PARENT directive in the zone file, except that relative domain names are treated as absolute names. By default, the parent's domain name is unspecified.
-po *`file`* OUTPUT=*`file`*	Specifies the name of the file to hold the bulk generated parent data. The default is FINISH-PARENT.

`/POLICY=`*`option`*

Specifies what policy to use when signing the zone. Specify one of the following options: (UNIX-syntax equivalents are also shown)

-dn DOWN	**DNSSIGNER** signs according to the DOWN policy. That is, the apex does not sign the parent's keys. The parent's keys and glue data are not expected from nor written to the parent files. This is the default.
-up UP	**DNSSIGNER** signs according to the UP policy. That is, the apex signs the parent's keys. The parent's keys and glue data are expected from and written to the parent files.

	This policy is not recommended.

-[no-]ess

/[NO]SELF_SIGN

/SELF_SIGN (-ess) instructs DNSSIGNER to make sure each key in the file is signed by its corresponding private key. This is done by implicitly adding $SIGNER directives to the zone file around each key set, adding those keys for just the set. If no private key is available, the $SIGNER directive remains in the output file.

The intent of this feature is to insert proof into DNS that the publickey's corresponding private key is held by the owner (or at least the entity signing the zone).

The default is **/NOSELF_SIGN (-no-ess)**.

/SIG=(*keyword*[,...])

Specifies options related to the generation of SIG RRs. Possible keywords (and their UNIX-style equivalents) are as follows.

For DURATION and PURGE_PERIOD, *ttl* format is taken from the BIND definition of TTL. Numeric seconds is accepted, as well as:

number W	weeks
number D	days
number H	hours
number M	minutes (not months!)
number S	seconds

The "end of the future" and "beginning of the past" are points in time which have the same time representation (one second apart) in a 32-bit roll-over specification of time. The end of the future is 2 to the 31st power seconds from the current time.

-dur *ttl* DURATION=*ttl*	All SIG records generated are set to expire at a the current time + duration. The default is 31 days.
-ks *domain algorithm keyid* [...] (to end of line)	There is no OpenVMS syntax equivalent. This adds the specified keys (key owner, algorithm, and key id) to the list of keys with which to sign. Equivalent to $SIGNER ADD <><> directives in the zone file. This switch is interpreted as the last switch of the command line. Any number of keys can be specified. See the description for **-k1** for the *domain* syntax.
-k1 *domain algorithm keyid* KEY=(DOMAIN=*domain*, ALGORITHM=*algorithm*, KEY_ID=*keyid*)	This adds the specified key (key owner, algorithm, and key id) to the list of keys with which to sign. Equivalent to a $SIGNER ADD <><> directive in the zone file. This switch may appear anywhere in the run command. It adds just one key. The default is that keys are specified by $SIGNER directives in the data files. A zone may elect not to use any keys.

	`domain` must be specified to match the key and private file names, with periods instead of dollar signs. This may not match the domain name if DNSKEYGEN had to abbreviate it to fit into an OpenVMS file name.

`-pt ttl`
`PURGE_PERIOD=ttl`
	Specifies that all SIG records with expiration times between the beginning of past up through (current time + the purge period) are treated as expired. SIG records with expiration times from (current + purge period) to the end of the future are retained if they are not proved invalid. The default is 1 week.

`-[no-]st`

`/[NO]STATISTICS`

/STATISTICS (-st) instructs DNSSIGNER to print summary statistics at the end of the run. The default is **/NOSTATISTICS (-no-st)**.

`/ZONE=(keyword[,...])`

Specifies options related to zone files. Possible keywords (and their UNIX-syntax equivalents) are as follows:

`-zi file`
`INPUT=file`
	Specifies the zone data input file. The first RR must be an SOA. The first record may be found in an `$INCLUDEd` file. The default is START-ZONE.

`-zo file`
`OUTPUT=file`
	Specifies the file where signed zone data is left. The default is FINISH-ZONE.

Examples

1. Assuming that the zone data is in f.zone and the parent file is in f.parent, to run the files through DNSSIGNER, do the following:

   ```
   $ IP dnssigner/zone=(input=f.zone)/parent=(input=f.parent)
   ```

   or

   ```
   $ dnssigner :== $IP$:dnssigner.exe
   $ dnssigner -zi f.zone -pi f.parent
   ```

   The outputs default to FINISH-ZONE and FINISH-PARENT. This does no signing, but merges the files, removes duplicates, generates NXT resource records, and makes signing instructions for them (if the zone is judged to be signed).

2. To sign the above zone with the key of test. key id 27782:

   ```
   $ IP dnssigner/zone=(input=f.zone)/parent=(input=f.parent) -
   /sig=(key=(domain=test.,alg=dsa,key_id=27782)
   ```

   or

   ```
   $ dnssigner -zi f.zone -pi f.parent -k1 test. dsa 27782
   ```

3. To sign with both keys 27782 and 3696:

```
   $ dnssigner -zi f.zone -pi f.parent -ks test. dsa 27782 test. dsa 3696
```

IP FONT COMPILE

IP FONT COMPILE — Compiles an ASCII BDF (bitmap distribution format) font file into a binary PCF (portable compiled format) file.

Format

`IP FONT COMPILE` [*qualifiers*] [*bdf_font_file*]

Qualifiers

/BIT_ORDER=*bit_order*

Specifies the order in which bits in each glyph are placed. Accepted values are MSBFIRST (most significant bit) or LSBFIRST (least significant bit).

The default is LSBFIRST on the OpenVMS AXP architecture.

/BYTE_ORDER=*byte_order*

Specifies the order in which multibyte data in the file is written. Multibyte data includes metrics and bitmaps. Accepted values are MSBFIRST (most significant bit) or LSBFIRST (least significant bit).

The default is LSBFIRST on the OpenVMS AXP architectures.

/OUTPUT=*file_name*

Specifies an output file name in which the results are written.

/PADDING=*font_glyph_padding*

Sets the font glyph padding. Each glyph in the font has each scanline padding into the specified size. Accepted values are BYTE, WORD, LONGWORD, or QUADWORD. On an OpenVMS AXP system the default is LONGWORD.

/SCANLINE=*data_size*

Specifies the unit of data swapped when the font bit order differs from the font byte order. Accepted values are BYTE, WORD, and LONGWORD. On an OpenVMS AXP system the default is LONGWORD.

/SERVER=*host:port*

The **/SERVER** qualifier specifies the server from which the font is read. The default value is LOCALHOST:7000.

IP FONT INFO

IP FONT INFO — Displays X font information useful for determining the capabilities and defined values of a font server.

Format

`IP FONT INFO` [*`qualifiers`*]

Qualifiers

`/OUTPUT=`*`file_name`*

Specifies an output file name in which the results are written.

`/SERVER=`*`host:port`*

Specifies the server from which the font is read (by default, LOCALHOST:7000).

IP FONT LIST

IP FONT LIST — Lists the font names that match a specified pattern.

Format

`IP FONT LIST` [*`qualifiers`*] [*`pattern`*]

Parameter

`pattern`

Specifies the pattern to match in font names. Wildcards are permitted in the patterns. If you do not specify a pattern, an asterisk (*) is assumed.

Qualifiers

`/BOUNDS`

Indicates long listings should display the minimum and maximum bounds of each font.

`/COLUMNS`

Indicates listings should display in multiple columns.

`/LISTING_TYPE=`*`size`*

Specifies the relative length of a font listing. Accepted values are SMALL, MEDIUM, LONG, and VERYLONG.

Consider using **/NOSORT** if you want LONG or VERYLONG listings faster; otherwise, these types of listings can take a long time to generate. You can also use **/OUTPUT** to write the results to a file.

`/NOSORT`

Indicates the listing is not sorted. Using this qualifier decreases the time required to produce a listing.

`/OUTPUT=`*`file_name`*

Specifies an output file name in which the results are written.

/SERVER=*host:port*

Specifies the server from which the font is read (by default, LOCALHOST:7000).

/WIDTH=*display_column_width*

Specifies the width of the columns (by default, 79).

IP FONT MKFONTDIR

IP FONT MKFONTDIR — Creates a `DECW$FONT_DIRECTORY.DAT` file in each specified directory.

Description

MKFONTDIR reads all font files in each specified directory. The order in which font files are read is `*.PCF` files, `*.SNF` files, then `*.BDF` files. For scalable fonts, you must edit the created `DECW$FONT_DIRECTORY.DAT` file to insert the X font name. If you edit this file, back up your changes so they are not lost when MKFONTDIR is run again.

The command fails if you do not have the necessary privileges to write into the directory you specify.

Format

IP FONT MKFONTDIR [*directory_name*]

Parameter

[*directory_name*]

Specifies the list of directories in which **IP FONT MKFONTDIR** creates a `DECW$FONT_DIRECTORY.DAT` file.

IP FONT SHOW

IP FONT SHOW — Displays font information from files that match the specified pattern.

Format

IP FONT SHOW [*qualifiers*] [*pattern*]

Parameter

[*pattern*]

Specifies the pattern to match in font names. Wildcards are permitted in the patterns. If you do not specify a pattern, an asterisk (*) is assumed.

Qualifiers

/BITMAP_PADDING=*bitmap_size*

Specifies how a character bitmap is padded. Accepted values are MINIMUM, MAXIMUM, and MAXWIDTH.

`/BIT_ORDER=`*`bit_order`*

Specifies the order in which bits in each glyph are placed. Accepted values are MSBFIRST (most significant bit) or LSBFIRST (least significant bit). The default is LSBFIRST on the OpenVMS AXP architectures.

`/BYTE_ORDER=`*`byte_order`*

Specifies the order in which multibyte data (including metrics and bitmaps) in the file is written. Accepted values are MSBFIRST (most significant bit) or LSBFIRST (least significant bit). The default is LSBFIRST on the OpenVMS AXP architectures.

`/END=`*`decimal_character_value`*

Specifies the ending character number (in decimal) about which you want font information listed. Use **/END** with the **/START** qualifier to specify character ranges. If you do not specify **/END**, all characters from the starting value to the end of the character set are listed. Possible values range from 0 to 255 for normal character sets, and from 0 to 65535 for X double-wide character sets.

`/EXTENTS`

Indicates that only the extents for a font are displayed.

`/OUTPUT=`*`file_name`*

Specifies an output file name in which the results are written.

`/PADDING=`*`font_glyph_padding`*

Sets the font glyph padding. Each glyph in the font has each scanline padding into the specified size. Accepted values are BYTE, WORD, LONGWORD, or QUADWORD. On an OpenVMS AXP system the default is LONGWORD.

`/SCANLINE=`*`data_size`*

Specifies the unit of data swapped when the font bit order differs from the font byte order. Accepted values are BYTE, WORD, and LONGWORD. On an OpenVMS AXP system the default is LONGWORD.

`/SERVER=`*`host:port`*

Specifies the server from which the font is read (by default, LOCALHOST:7000).

`/START=`*`decimal_character_value`*

Specifies the starting character number (in decimal) about which you want font information listed. Use **/START** with the **/END** qualifier to specify character ranges. If you do not specify **/END**, all characters from the starting value to the end of the character set are listed. Possible values range from 0 to 255 for normal character sets, and from 0 to 65535 for X double-wide character sets.

IP FONT UNCOMPILE

IP FONT UNCOMPILE — Converts a binary PCF-format font file to an ASCII BDF-format file.

Format

`IP FONT UNCOMPILE` [*`qualifiers`*] [*`pcf_font_file`*]

Qualifiers

`/OUTPUT=`*`file_name`*

Specifies the output file name into which the results are written.

`/SERVER=host:port`

Specifies the server from which the font is read (by default, LOCALHOST:7000).

Example

```
$ IP FONT UNCOMPILE -
_$ -Adobe-Helvetica-Medium-R-Normal--25-180-100-100-P-130-ISO8859-1
```

IP GATED /CHECK

IP GATED /CHECK — Checks the syntax of a GateD configuration file. If no input file is specified, VSI TCP/IP checks the default configuration file, `IP$:GATED.CONF`. This command does not affect a running GateD process.

Format

`IP GATED /CHECK`

Parameter

`filename`

Name of the configuration file to check. If omitted, defaults to `IP$:GATED.CONF`.

Example

Checks the syntax of a GateD configuration file called `TEST.CONF` located in the user's current working directory.

```
$ IP GATED/CHECK TEST.CONF
```

IP GATED /DUMP

IP GATED /DUMP — Tells GateD to dump its internal state into a text file. If you omit the filename, the default is `IP$:GATED.DUMP`.

Format

`IP GATED /DUMP` [log]

Parameter

`log`

Contains log statements generated by GateD. If omitted, defaults to `IP$:GATED.DUMP`.

IP GATED /LOAD

IP GATED /LOAD — Tells the GateD process to load a configuration file. If no file is specified, the default file `IP$:GATED.CONF` is loaded. If the GateD process detects an error in the configuration file being loaded, it stops running.

Format

`IP GATED /LOAD` [*`file`*]

Parameter

`file`

Name of the configuration file to load. If omitted, defaults to `IP$:GATED.CONF`.

Example

This example tells the GateD process to load a new configuration file called `TEST_CONFIG.CONF` from the system manager’s current working directory.

```
$ IP GATED/LOAD TEST_CONFIG.CONF
```

IP GATED /SET /TRACE

IP GATED /SET /TRACE — Tells the GateD process to turn on or off various tracing flags. This controls what is placed in the `IP$:GATED.LOG` file. By default, minimal tracing is done.

Format

`IP GATED /SET /TRACE` *`qualifier`*

Qualifiers

`/ALL`

Turns on all tracing.

`/[NO]DETAILS`

Sets tracing of all send and receive information.

`/[NO]RECV_DETAILS`

Sets tracing of receive information.

`/[NO]SEND_DETAILS`

Sets tracing of send information.

`/[NO]EVENTS`

Sets tracing of normal events.

`/NONE`

Turns off all tracing.

`/[NO]PACKETS`

Sets tracing of packet sends and receives.

`/[NO]RECV_PACKETS`

Sets tracing of packet receives.

`/[NO]SEND_PACKETS`

Sets tracing of packet sends.

`/[NO]PARSING`

Sets tracing of configuration file parsing.

`/[NO]POLICY`

Sets tracing of policy decisions.

`/[NO]ROUTING`

Sets tracing of routing table changes.

`/[NO]STATES`

Sets tracing of state machine transitions.

`/[NO]SYMBOLS`

Sets tracing of kernel symbols.

`/[NO]TASKS`

Sets tracing of task and job functions.

`/[NO]TIMER`

Sets tracing of timer functions.

Example

This example tells the GateD process to turn on tracing of policy decisions and turn off tracing of state machine transitions.

```
$ IP GATED/SET/TRACE/POLICY/NOSTATES
```

IP GATED /SHOW /OSPF

IP GATED /SHOW /OSPF — Queries OSPF routers. You can obtain a wide variety of detailed information from these routers using these commands.

Format

```
IP GATED /SHOW /OSPF  option
```

Description

All of the **SHOW OSPF** commands use a file called `IP$:OSPF_DESTS.DAT`. This is a file of OSPF destination records. Each record is a single line entry listing the destination IP address, the destination host name, and an optional OSPF authentication key (if the destination activates authentication).

Caution

Since the `OSPF_DESTS.DAT` file may contain authentication information, you should restrict access to it.

To stop the output of this command, enter a **Ctrl**/**C** at the command line.

Options

`/ADVERTISE ADVERTISE` *`area-id type ls-id adv-router index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

Displays link state advertisements. The parameters and qualifiers for **IP SHOW OSPF ADVERTISE** are as follows:

Parameter and Qualifier	Description
`area-id`	OSPF area for which the query is directed.
`type`	The available types are **/INTERFACES** — Requests the router links advertisements. Describes the collected states of the router's interfaces. For this request, the ls-id field should be set to the originating router's Router ID. **/ROUTERS** — Requests the network links advertisements. Describes the set of routers attached to the network. For this request, the ls-id field should be set to the IP interface address of the network's Designated Router. **/NETWORK_ROUTES** — Requests the summary link advertisements describing routes to networks. Describes the inter-

Parameter and Qualifier	Description
	area routes and enables the condensing of routing information at area borders. For this request, the ls-id field should be set to the destination network's IP address. **/BOUNDARY_ROUTES** — Requests the summary link advertisements describing routes to AS boundary routers. Describes the inter-area routes and enables the condensing of routing information at area borders. For this request, the ls-id field should be set to the Router ID of the described AS boundary router. **/EXTERNAL_ROUTES** — Requests the AS external link advertisements. Describes routes to destinations external to the AS. For this request, the ls-id field should be set to the destination network's IP address.
`ls-id`	See the type parameter.
`adv-route`	Router ID of the router that originated this link state advertisement.
`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=file`	Name of an output file to write the results to.
`/FILE=file`	Alternate file of OSPF destination records to use.
`/TIMEOUT=seconds`	Interval to wait for a response. Default is 20 seconds.

`/AS AS index`

`/OUTPUT=file`

`/FILE=file`

`/TIMEOUT=seconds`

Shows the Autonomous System (AS) external database entries. This table reports the advertising router, forwarding address, age, length, sequence number, and metric for each AS external route. The parameters and qualifiers for **IP GATED/ SHOW/OSPF/AS** are as follows:

`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=file`	Name of an output file to write the results to.
`/FILE=file`	Alternate file of OSPF destination records to use.
`/TIMEOUT=seconds`	Interval to wait for a response. Default is 20 seconds.

`DESTINATIONS/OUTPUT=file`

`/FILE=file`

This command displays the list of destinations and their indices described in an OSPF destination records file. The parameters and qualifiers for **IP GATED/SHOW/OSPF/ DESTINATIONS** are as follows:

`/OUTPUT=file`	Name of an output file to write the results to.
`/FILE=file`	Alternate file of OSPF destination records to use.

`/ERRORS ERRORS` *`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

Shows the error log. This reports the different error conditions that can happen between OSPF routing neighbors and shows the number of occurrences for each. The parameters and qualifiers for **IP GATED/SHOW/OSPF/ERRORS** are as follows:

index	Indexes into a file of OSPF destination records.
/OUTPUT=*file*	Name of an output file to write the results to.
/FILE=*file*	Alternate file of OSPF destination records to use.
/TIMEOUT=*seconds*	Interval to wait for a response. Default is 20 seconds.

`/HOPS HOPS` *`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

Shows the set of next hops for the OSPF router being queried. The parameters and qualifiers for **IP GATED/SHOW/OSPF/HOPS** are as follows:

index	Indexes into a file of OSPF destination records.
/OUTPUT=*file*	Name of an output file to write the results to.
/FILE=*file*	Alternate file of OSPF destination records to use.
/TIMEOUT=*seconds*	Interval to wait for a response. Default is 20 seconds.

`/INTERFACES INTERFACES` *`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

Displays all interfaces. This shows all the interfaces configured for OSPF. The information includes the area, interface IP address, interface type, interface state, cost, priority and the IP address of the DR and BDR of the network. The parameters and qualifiers for **IP GATED SHOW OSPF INTERFACES** are as follows:

index	Indexes into a file of OSPF destination records.
/OUTPUT=*file*	Name of an output file to write the results to.
/FILE=*file*	Alternate file of OSPF destination records to use.
/TIMEOUT=*seconds*	Interval to wait for a response. Default is 20 seconds.

`/LOG LOG` *`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

Shows the cumulative log. This log includes input and output statistics for monitor requests, hellos, database descriptions, link state updates, and link state ACK packets. Area statistics are provided that describe the total number of routing neighbors and number of active OSPF interfaces. Routing table statistics are summarized and reported as the number of intra-area routes, inter-area routes, and AS external database entries.

The parameters and qualifiers for **IP GATED/SHOW/OSPF/LOG** are as follows:

`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=`*`file`*	Name of an output file to write the results to.
`/FILE=`*`file`*	Alternate file of OSPF destination records to use.
`/TIMEOUT=`*`seconds`*	Interval to wait for a response. Default is 20 seconds.

`/NEIGHBORS NEIGHBORS `*`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

`/RETRANSMIT`

This command shows all OSPF routing neighbors. The information shown includes the area, local interface address, router ID, neighbor IP address, state and mode. The parameters and qualifiers for **IP GATED/SHOW/OSPF/NEIGHBORS** are as follows:

`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=`*`file`*	Name of an output file to write the results to.
`/FILE=`*`file`*	Alternate file of OSPF destination records to use.
`/TIMEOUT=`*`seconds`*	Interval to wait for a response. Default is 20 seconds.
`/RETRANSMIT`	Displays the retransmit list of neighbors.

`/ROUTING ROUTING `*`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

Shows the OSPF routing table. This table reports the AS border routes, area border routes, summary AS border routes, and the networks managed using OSPF. The parameters and qualifiers for **IP GATED/SHOW/OSPF/ROUTING** are as follows:

`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=`*`file`*	Name of an output file to write the results to.

`/FILE=`*`file`*	Alternate file of OSPF destination records to use.
`/TIMEOUT=`*`seconds`*	Interval to wait for a response. Default is 20 seconds.

`/STATE STATE` *`index`*

`/OUTPUT=`*`file`*

`/FILE=`*`file`*

`/TIMEOUT=`*`seconds`*

`/RETRANSMIT`

Shows the link state database (except for ASEs). This describes the routers and networks making up the AS. The parameters and qualifiers for **IP GATED/SHOW/OSPF/STATE** are as follows:

`index`	Indexes into a file of OSPF destination records.
`/OUTPUT=`*`file`*	Name of an output file to write the results to.
`/FILE=`*`file`*	Alternate file of OSPF destination records to use.
`/TIMEOUT=`*`seconds`*	Interval to wait for a response. Default is 20 seconds.
`/RETRANSMIT`	Displays the retransmit link state database.

Examples

1. Displays the OSPF cumulative log for index 1 in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/LOG 1
  Source <<192.168.5.31 izar.nene.com>>
IO stats
  Input   Output   Type
  2       0        Monitor request
  0       0        Hello
  0       0        DB Description
  0       0        Link-State Req
  0       0        Link-State Update
  0       0        Link-State Ack
  ASE:    0 checksum sum 0

  LSAs originated: 39  received: 0
  Router: 39

  Area 0.0.0.0:
  Neighbors: 0  Interfaces: 0
  Spf: 1 Checksum sum CE9D
  DB: rtr: 1 net: 0 sumasb: 0 sumnet: 0

Routing Table:
        Intra Area: 0   Inter Area: 0   ASE: 0
```

2. Displays the OSPF interface log for index 1 in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/INTERFACE 1
  Source <<192.168.5.31      izar.nene.com>>
IO stats
  Input   Output   Type
  6       0        Monitor request
  0       0        Hello
  0       0        DB Description
  0       0        Link-State Req
  0       0        Link-State Update
  0       0        Link-State Ack
  ASE: 0 checksum sum 0

  LSAs originated: 39  received: 0
  Router: 39

  Area 0.0.0.0:
  Neighbors: 0  Interfaces: 0
  Spf: 1  Checksum sum CE9D
  DB: rtr: 1 net: 0 sumasb: 0  sumnet: 0

Routing Table:
        Intra Area: 0   Inter Area: 0   ASE: 0
```

3. Displays the OSPF destination records in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/DESTINATIONS
1: 192.168.5.31    izar.nene.com
```

4. Displays the OSPF link state database log for index 1 in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/STATE 1
          Source <<192.168.5.31    izar.nene.com>>
LS Data Base:
Area: 0.0.0.0
Type LinkState ID   AdvRouter          Age   Len Sequence Metric  Where
-------------------------------------------------------------------------
Rtr  192.168.5.31   192.168.5.31 986   24    80000027 0
 SpfTree
```

5. Displays the OSPF next hops log for index 1 in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/HOPS 1
          Source <<192.168.5.31    izar.nene.com>>
Next hops:

Address        Type       Refcount       Interface
---------------------------------------------------------
192.168.5.31   Direct 1   192.168.5.31   SVA-0
```

6. Displays the OSPF error log for index 1 in the `OSPF_DESTS.DAT` file.

```
$ IP GATED/SHOW/OSPF/ERRORS 1
  Source <<192.168.5.31  izar.nene.com>>
Packets Received:
 3: Monitor request                  0: Hello
 0: DB Description                   0: Link-State Req
 0: Link-State Update                0: Link-State Ack

Packets Sent:
 0: Monitor response                 0: Hello
 0: DB Description                   0: Link-State Req
 0: Link-State Update                0: Link-State Ack
Errors:
 0: IP$: bad destination           0: IP$: bad protocol
 0: IP$: received my own packet    0: OSPF: bad packet type
 0: OSPF: bad version              0: OSPF: bad checksum
 0: OSPF: bad area id              0: OSPF: area mismatch
 0: OSPF: bad virtual link         0: OSPF: bad authentication type
 0: OSPF: bad authentication key   0: OSPF: packet too small
 0: OSPF:packet size > ip length   0: OSPF: transmit error
 0: OSPF: interface down           0: OSPF: unknown neighbor
 0: HELLO: netmask mismatch        0: HELLO: hello timer mismatch
 0: HELLO: dead timer mismatch     0: HELLO: extern option mismatch
 0: HELLO: router id confusion     0: HELLO: virtual neighbor unknown
 0: HELLO: NBMA neighbor unknown   0: DD: neighbor state low
 0: DD: router id confusion        0: DD: externoption mismatch
 0: DD: unknown LSA type           0: LS ACK: neighbor state low
 0: LS ACK: bad ack                0: LS ACK: duplicate ack
 0: LS ACK: Unknown LSA type       0: LS REQ: neighbor state low
 0: LS REQ: empty request          0: LS REQ: bad request
 0: LS UPD: neighbor state low     0: LS UPD: newer self-gen LSA
 0: LS UPD: LSA checksum bad       0: LS UPD:received less recent LSA
 0: LS UPD: unknown LSA type
```

IP GATED /SHOW /RIP

IP GATED /SHOW /RIP — Used to request all routes known by a RIP gateway. The routing information in any routing packets returned is displayed numerically and symbolically. This command is intended to be used as a tool for debugging gateways, not for network management. To stop the output of this command, enter a **Ctrl**/**C** at the command line.

Format

`IP GATED /SHOW /RIP` *`gateway-ia`*

Parameter

`gateway-ia`

Internet address or name of the gateway to be queried.

Qualifiers

`/AUTHENTICATION=`*`authkey`*

Authentication password to use for queries. If specified, an authentication type of SIMPLE is used. The default authentication type is NONE.

`/NONAME`

Prevents the responding host's address from being looked up to determine the symbolic name.

`/POLL`

Requests information from the gateway's routing table. This is the default. If there is no response to the **/POLL** qualifier, the **/REQUEST** qualifier is tried.

`/REQUEST`

Requests information from the gateway's routing table. Unlike the **/POLL** qualifier, all gateways should support this command. If there is no response, the **/POLL** qualifier is tried.

`/TIMEOUT=`*`seconds`*

Number of seconds to wait for the initial response from a gateway. Default is 5 seconds.

`/TRACE`

Traces the RIP packets being sent and received by this command.

`/V1`

Sends the query as a RIP version 1 packet.

`/V2`

Sends the query as a RIP version 2 packet.

Example

Shows the routers known by RIP gateway 192.168.10.2.

```
$ IP GATED/SHOW/RIP 192.168.10.2
24 bytes from omega1.foobar.com(192.168.10.2):
  net/mask                    router         metric   tag
  192.168.5.0/255.255.255.0   192.168.10.1   2        0000
```

IP GATED /SHOW /TRACE

IP GATED /SHOW /TRACE — Queries tracing in GateD.

Format

```
IP GATED /SHOW /TRACE
```

Example

```
$ IP gated/show/trace
 Summary of GateD tracing
-------------------------------------------------------
 State Machine Transitions Logging    is :  'OFF'
 Internal Events Logging              is :  'OFF'
 Policy Decision Logging              is :  'OFF'
 Task Information Logging             is :  'OFF'
 Timer Logging                        is :  'OFF'
 Routing Information Logging          is :  'OFF'
 General Send and Receive Logging     is :  'OFF'
 General Receive Logging              is :  'OFF'
 General Send Logging                 is :  'OFF'
 Packet Send and Receive Logging      is :  'OFF'
 Packet Receive Logging               is :  'OFF'
 Packet Send Logging                  is :  'OFF'
 Configuration File Parsing Logging   is :  'OFF'
 Route Advertisement Logging          is :  'OFF'
 Kernel Symbols Logging               is :  'OFF'
 Network Interface Logging            is :  'OFF
```

IP GATED /STOP

IP GATED /STOP — Tells the GateD process to halt in an orderly manner.

Format

```
IP GATED /STOP
```

IP GATED /TOGGLE_TRACING

IP GATED /TOGGLE_TRACING — Toggles GateD tracing on and off. This command opens and closes the GateD log file `IP$:GATED.LOG` as needed.

Format

```
IP GATED /TOGGLE_TRACING
```

IP GATED /UPDATE_INTERFACES

IP GATED /UPDATE_INTERFACES — Tells the GateD process to rescan the network interfaces.

Format

```
IP GATED /UPDATE_INTERFACES
```

IP HOST_TABLE COMPILE

IP HOST_TABLE COMPILE — The VSI TCP/IP host table compiler generates binary host tables from the ASCII host table files. After modifying a VSI TCP/IP host table, use this command to compile it into its binary form.

Format

IP HOST_TABLE COMPILE[*files*]

Description

After recompiling your host tables, reinstall the host tables by rebooting, or by invoking the **@IP $:INSTALL_DATABASES** command. Then make the host table usable to the IP$SERVER process servers by restarting this process with the **@IP$:START_SERVER** command. VSI TCP/IP uses the compiled host tables for fast lookups of host names, and for translation of host, network, protocol, and service names to numbers.

Parameter

`files`

Contains a comma-separated list of one or more input files to be compiled. These files must be in the format described in RFC-952 "DoD Internet Host Table Specification." If not specified, the input files default to `IP$:HOSTS.SERVICES`, `IP$:HOSTS.LOCAL`, and `IP$:HOSTS.TXT`.

Qualifiers

`/HOST_TABLE_FILE=`*`file`*

Specifies the file to which the compiler writes the binary host table (by default, `IP $:NETWORK_DATABASE`).

`/SILENTLY`

`/NOSILENTY (default)`

Determines whether the compilation proceeds quietly. The default, **/NOSILENTLY**, can take some time to process.

`/STARTING_HASH_VALUE=`*`value`*

Specifies the initial hash size for the host table hash. Starting at this value, the host table compiler searches for an acceptable hashing function. The default for this qualifier is the "best value," which is computed from the size of the data as the utility attempts to create 512-byte units.

When you run **HOST TABLE COMPILE**, the hash value is listed in the displayed messages. To select a value for this qualifier, choose a number from the displayed range of values.

`/TBLUK_FILE=`*`file`*

Specifies the file to which the compiler writes the "host-completion" database, used by programs that allow for escape-completion of partially typed host names. The default is `IP$:HOSTTBLUK.DAT`.

`/UNIX_HOST_FILE=`*`file`*

Specifies the file to which the compiler writes a UNIX-style hosts file that can be used on most UNIX systems and with many other vendors' TCP implementations. The default **/NOUNIX_HOST_FILE**, inhibits the creation of a UNIX-style hosts file.

IP HOST_TABLE GET

IP HOST_TABLE GET — Connects to the HOSTNAME port of the host you specify, and uses the HOSTNAME protocol to retrieve the `HOSTS.TXT` file. After retrieving a new VSI TCP/IP host table, compile it into binary form with the **IP HOST_TABLE COMPILE** command so the host table can be accessed.

Qualifiers

`/HOST=`*`host`*` (default NIC.DDN.MIL)`

Specifies a host other than `NIC.DDN.MIL`. If you specify the host name instead of the address, the host name must exist in your existing host tables.

/OUTPUT_FILE=*file*

Specifies a different output file (by default, `IP$:HOSTS.TXT`).

/SILENTLY

/NOSILENTLY (default)

Specifies that various debugging information is written to `SYS$ERROR` as the program executes.

/QUERY

Specifies an arbitrary HOSTNAME protocol request to the host of interest as follows:

- If the **/QUERY** qualifier is present, use its value
- Otherwise, if the **/VERSION** qualifier is present, user "VERSION"
- Otherwise, use "ALL"

/VERSION

Retrieves only the `HOSTS.TXT` version number.

The HOSTNAME protocol supports simple text query requests of the form:

command_key argument(s) [*options*]

command_key is a keyword indicating the nature of the request and square brackets ([ ]) indicate an optional field. The defined keys are described in the following table:

Keyword	**Response**
HELP	The information in this table.
VERSION	"VERSION: *string*" where *string* is different for each version of the host table.
HNAME *hostname*	One or more matching host table entries.
HADDR *hostaddr*	One or more matching host table entries.
ALL	The entire host table.
ALL-OLD	The entire host table without domain-style names.
DOMAINS	The entire top-level domain table (domains only).
ALL-DOM	Both the entire domain table and the host table.
ALL-INGWAY	All known gateways in TENEX/TOPS-20 INTERNET.GATEWAYS format.

IP HOST_TABLE INSTALL

IP HOST_TABLE INSTALL — Installs the binary host tables as global sections. Do not run **HOST_TABLE INSTALL** directly. Instead, use the **IP$:INSTALL_DATABASES.COM** command procedure.

Format

```
IP HOST_TABLE INSTALL
```

IP IPP SHOW

IP IPP SHOW — The **IP IPP SHOW** utility allows a user to learn the capabilities supported by an IPP server. This utility queries the server and displays the supported attributes. The program can be used to see what a given server supports, by a program to gather information about a number of printers, or by a DCL or other program to check the capabilities of a given server before submitting a print job to a queue. The command syntax is: $ **IP IPP SHOW** ***server_URI*** **/qualifiers...**.

Qualifiers

/ATTRIBUTE=*attribute*

Puts the program into a mode suitable for use from a DCL command procedure. Not compatible with the **/FORMAT** or **/OUTPUT** qualifiers or those associated with them. It causes the program to return the value of a single attribute as a character string in a DCL symbol. The symbol may be specified with the **/SYMBOL** qualifier if the default of “IPP_SHOW_RESULT” is not desired. This is intended for use in a procedure to check to see if, for example, a given server supports color printing before submitting a job to a queue that requires color output. Allowable values for *attribute* are:

Charset_Configured	Orientation_Requested_Default
Charset_Supported	Orientation_Requested_Supported
Color_Supported	Page_Ranges_Default
Compression_Supported	Page_Ranges_Supported
Copies_Default	PDL_Override_Supported
Copies_Supported	Print_Quality_Default
Document_Format_Default	Print_Quality_Supported
Document_Format_Supported	Printer_Current_Time
Finishings_Default	Printer_Driver_Installer
Finishings_Supported	Printer_Info
Gen_Natural_Language_Supported	Printer_Is_Accepting_Jobs
Job_Hold_Until_Default	Printer_Location
Job_Hold_Until_Supported	Printer_Make_and_Model
Job_Impressions_Supported	Printer_Message_From_Operator
Job_K_Octets_Supported	Printer_More_Info
Job_Media_Sheets_Supported	Printer_More_Info_Manufacturer

Job_Priority_Default	Printer_Name
Job_Priority_Supported	Printer_Resolution_Default
Job_Sheets_Default	Printer_Resolution_Supported
Job_Sheets_Supported	Printer_State
Media_Default	Printer_State_Message
Media_Supported	Printer_State_Reasons
Multiple_Doc_Handling_Default	Printer_Uptime
Multiple_Doc_Handling_Supported	Printer_URI_Supported
Multiple_Operation_Timeout	Queued_Job_Count
Natural_Language_Configured	Reference_URI_Schemes_Supported
Number_Up_Default	Sides_Default
Number_Up_Supported	Sides_Supported
Operations_Supported	URI_Security_Supported

/[NO]APPEND

Specifies that output should be appended to an existing output file if possible. **/NOAPPEND** is the default.

/FORMAT=*style*

Specifies what print style to use. *style* is either

- "SCREEN" (default) which writes in a human-friendly screen-formatted mode or
- "LIST" which writes an easy to parse, name=value format, one name/value pair per line.

/[NO]FULL

Causes all IPP attributes to be included in the display, whether the server supports them or not. Those not supported are marked as such. **/NOFULL** is the default.

/[NO]GLOBAL

Specifies whether the named symbol should be created as a DCL global symbol. Used only with **/ATTRIBUTE**. If specified as **/NOGLOBAL**, the symbol will be local to the calling procedure level. **/GLOBAL** is the default.

/OUTPUT=*file*

Specifies a file to write output to. "SYS$OUTPUT:" is the default.

/SYMBOL=*symbolname*

Specifies a DCL symbol name that should be set to the value of the specified attribute. Used only with **/ATTRIBUTE**. The default is "IPP_SHOW_RESULT" if **/SYMBOL** is not specified.

Examples of **IP IPP SHOW** Use and Output

1. Basic operation with all defaults:

```
$ IP IPP SHOW LILLIES.FLOWERPOTS.COM
LILLIES.FLOWERPOTS.COM as of Tue Mar 14 16:08:43 2017

CURRENT INFO:
 Printer State:      Idle
 State Reasons:      none
 Accepting Jobs?:    Yes
 Queued Job Count:   0

PRINTER INFO:
 Name:               Lexmark Optra T610
 Make & Model:       Lexmark Optra T610

DEFAULTS:
 Document Format:    application/octet-stream
 Orientation:        Portrait
 Number-Up:          1
 Copies:             1
 Job Media Sheets:   none
 Character Set:      utf-8
 Natural Language:   en-us

SUPPORTED FEATURES AND ALLOWED VALUES:
 Color?:              No
 Orientation:         Portrait, Landscape
 Document Formats:    application/octet-stream, application/postscript,
                      application/vnd.hp-PCL, text/plain
 Job Sheets:          none, standard
 Number-Up:           1:16
 Copies:              1:999
 PDL Override:        not-attempted
 Character Sets:      utf-8, us-ascii
 Natural Languages:   en-us
 Operations:          Print_Job, Validate-Job, Cancel-Job,
                      Get-Job_Attributes, Get-Jobs,
                      Get-Printer_Atrributes, Unknown: 18

URIs Supported and associated security options:
 URI:  http://192.168.50.2/
 Security: none

 URI:  http://192.168.50.2:631/
 Security: none
```

2. Operation with /**FULL** and output to a file (note that the "/" character in the URI requires use of quotes around the server URI parameter):

```
$ IP IPP SHOW "LILLIES.FLOWERPOTS.COM/IPP" /FULL /OUTPUT=FOO.BAR
FOO.BAR contains:

LILLIES.FLOWERPOTS.COM/IPP as of Tue Mar 14 16:11:54 2017

CURRENT INFO:
 Printer State:      Idle
```

```
   State Reasons:      none
   State Message:      <not supported>
   Accepting Jobs?:    Yes
   Queued Job Count:   <not supported>
   Uptime (seconds):   <not supported>
   Printer Time:       <not supported>

  PRINTER INFO:
   Name:                LILLIES
   Printer Location:    <not supported>
   Printer Info:        MANUFACTURER:Hewlett-Packard;COMMAND SET:PJL,ML -
                        C,PCL,PCLXL,POSTSCRIPT;MODEL:HP LaserJet 2100  -
                        Series;CLASS:PRINTER;DESCRIPTION:H
   URL for more info:   <not supported>
   URL for driver:      <not supported>
   Make & Model:        <not supported>
   URL for Maker:       <not supported>

  DEFAULTS:
   Document Format:     application/octet-stream
   Orientation:         <not supported>
   Number-Up:           <not supported>
   Sides:               <not supported>
   Copies:              <not supported>
   Mult. Doc. Handling: <not supported>
   Media:               <not supported>
   Job Media Sheets:    <not supported>
   Finishings:          <not supported>
   Job Priority:        <not supported>
   Job Hold Until:      <not supported>
   Print Quality:       <not supported>
   Printer Resolution:  <not supported>
   Character Set:       us-ascii
   Natural Language:    en-us
   Mult. Op. Timout:    <not supported>

  SUPPORTED FEATURES AND ALLOWED VALUES:
   Color?:              <not supported>
   Orientation:         <not supported>
   Document Formats:    text/plain, text/plain; charset=US-ASCII,
                        application/postscript, application/vnd.hp-PCL,
                        application/octet-stream
   Job Sheets:          <not supported>
   Number-Up:           <not supported>
   Sides:               <not supported>
   Copies:              <not supported>
   Mult. Doc. Handling: <not supported>
   Media Names:         <not supported>
   Job Media Sheets:    <not supported>
   Finishings:          <not supported>
   Job Priority:        <not supported>
   Job Hold Until:      <not supported>
   Page Ranges?:        <not supported>
   Print Qualities:     <not supported>
   Resolutions:         <not supported>
   Compression Modes:   <not supported>
   Job K-octets:        <not supported>
   Job Impressions:     <not supported>
```

```
    PDL Override:        not-attempted
    Character Sets:      us-ascii, utf-8
    Natural Languages:   en-us
    URI Schemes:         <not supported>
    Operations:          Print_Job, Validate-Job, Cancel-Job,
                         Get-Job_Attributes, Get-Jobs,
                         Get-Printer_Atrributes

   URIs Supported and associated security options:

     URI:          /ipp
     Security:     none

     URI:          /ipp/port1
     Security:     none

   MESSAGE FROM OPERATOR:

    <no Message>
```

3. Operation with /attribute and **/SYMBOL** and **/GLOBAL** to get a single attribute into a DCL symbol:

```
$ MULT IPP SHOW LEXIM /ATTRIB=NUMBER_UP_SUPPORTED /SYMBOL=NUMUP /GLOBAL
$ SHO SYM NUMUP
  NUMUP == "1:16"
$
```

IP LOAD

IP LOAD — Invokes the VSI TCP/IP network LOADER. This program loads a network image into the OpenVMS kernel and starts the network. This utility is invoked automatically by the network startup command file generated by the Network Configuration Utility and should not be invoked by a user.

Format

```
IP LOAD
```

IP NETCONTROL

IP NETCONTROL — Sends commands to services internal to the IP$SERVER process. **NETCONTROL** can select any server provided in the VSI TCP/IP configuration or those previously added with the Server Configuration Utility. This command affects only the currently running configuration.

Format

`IP NETCONTROL` [*service*] [*command*]

Parameters

[*service*]

Connects to the specified service (by default, **NETCONTROL**).

[*command*]

Sends a specified command string to the server. If you do not specify a command string, **NETCONTROL** enters interactive mode.

Qualifiers

/HOST=*host*

Connects to the **NETCONTROL** service on the specified host (by default, the **NETCONTROL** service on the local host).

/VERBOSE

Displays the entire **NETCONTROL** protocol conversation. This qualifier is useful only for debugging purposes.

Description

The **NETCONTROL** program sends commands to services internal to the IP$SERVER process. **NETCONTROL** currently provides access to the following VSI TCP/IP services:

Table 1.2. VSI TCP/IP NETCONTROL Services

ACCESS	BOOTP	BWNFSD	CLUSTERALIAS
DHCLIENT	DHCP	DOMAINNAME	EKLOGIN
FONTSERVER	GATED	IPXRIP	KERBEROS
KLOGIN	KSHELL	NETCONTROL	NFS
NFSV3	NOT	NTP	PCNFSD
RACOON	RARP	“R” Server	RDISC
REXEC	RLOGIN	RPCBOOTPARAMS	RPCLOCKMGR
RPCMOUNT	RPCPORTMAP	RPCQUOTAD	RPCSTATUS
RSHELL	SAP	SNMP	SSH
SYSLOG	TELNET	TFTP	UCXQIO
VIADECNET	VIAPSI	XDM	

For loadable services (those with an INIT setting of Merge_image), you can use the **SERVER-CONFIG SET PROCESS *process_name*** command to have the service run in an auxiliary master server process with the specified name, rather than in the main master server process (which has a process name of IP$SERVER).

Use the SERVER-CONFIG **SET FLAGS START_AUX_SERVER** command to have the main master server start the auxiliary server process automatically.

Services running in auxiliary master server processes can be controlled with **NETCONTROL**, independent of whether the START_AUX_SERVER flag is set, or whether the auxiliary process was started manually.

Auxiliary server processes are most useful for services which may require very large quantities of process quotas such as virtual memory. They can also be useful when there are problems with a loadable service that cause the master server to terminate abnormally; the service can be isolated in a separate process to prevent other services from being interrupted when the abnormal termination occurs.

The following command shuts down all services, including those run in auxiliary master server processes. (In earlier versions, services running in auxiliary master server processes were not affected by this command.)

```
$ IP NETCONTROL NETCONTROL SHUTDOWN ALL
```

In addition, specifying a **SET PROCESS** command on the **NETCONTROL** service (in SERVER-CONFIG) has no effect; the **NETCONTROL** server always runs in the main master server process.

Invoke **NETCONTROL** with **IP NETCONTROL**, or with **NETCONTROL** commands from inside the Server Configuration Utility or NFS Server Configuration Utility.

NETCONTROL connects to the **NETCONTROL** server on the local host, or on a remote host if one is specified. The following example demonstrates two ways of connecting to the RLOGIN service. Note: the **NETCONTROL** prompt indicates the name of the service to which you are connected.

```
$ IP NETCONTROL RLOGIN
Connected to NETCONTROL server on "LOCALHOST" < SP1.SPROCKET.COM Network
 Control 10.5 (nnn) at Mon 13-Mar-2017 7:42am-EST
RLOGIN>
$ IP NETCONTROL
Connected to NETCONTROL server on "LOCALHOST"
< SP1.SPROCKET.COM Network Control 10.5 (nnn) at Mon 13-Mar-2017 7:42am-EST
NETCONTROL>SELECT RLOGIN
RLOGIN>
```

The following example shows how to specify a **NETCONTROL** command from the command line. When used this way, **NETCONTROL** exits to DCL upon completion.

```
$ IP NETCONTROL RLOGIN SHOW
Connected to NETCONTROL server on "LOCALHOST"
< SP1.SPROCKET.COM Network Control 10.5 (nnn) at Mon 13-Mar-2017 7:42am-EST
 < File Cache:
< IP$:HOSTS.EQUIV (Expires in 59 minutes)
< USERS:[MIGUEL].rhosts (Expired)
< Authorization Cache:
< EDUARDA (Expires in 59 minutes)
< MIGUEL  (Expired)
$
```

You can also use **NETCONTROL** to control the IP$SERVER on a remote system, subject to the restrictions set on that system's **NETCONTROL** server.

The following example shows how to invoke **NETCONTROL** on a remote system.

```
$ IP NETCONTROL/HOST=SP1.SPROCKET.COM RLOGIN
Connected to NETCONTROL server on "SP1.SPROCKET.COM"
< SP1.SPROCKET.COM Network Control 10.5 (nnn) at Mon 13-Mar-2017 7:42am-EST
RLOGIN>
```

To change the ACCOUNTING and *DEBUG* parameters with the **IP NETCONTROL** command, use the following commands. Use the *DEBUG* parameter to dynamically set the VSI TCP/IP server debugging level to the specified value *n*. By default, additional information is provided in the accounting record by the VSI TCP/IP server. You can disable this feature by setting *n* to 0. When set to 1, the remote name and service name are added to the ACCOUNTING record.

```
$ IP NETCONTROL NETCONTROL DEBUG n
$ IP NETCONTROL NETCONTROL ACCOUNTING n
```

Table 1.3 shows the NETCONTROL commands you can use at any time.

Table 1.3. NETCONTROL Commands Valid at any Time

Command	Description
`LIST`	Prints a list of the active services that support **NETCONTROL**.
`NOOP`	Does nothing; provided for testing the **NETCONTROL** server.
`QUIT`	Exits **NETCONTROL**.
`QUOTE` *`string`*	Sends a string verbatim to the **NETCONTROL** server.
`SELECT` *`service`*	Selects a service on which to operate. Use the question mark (?) character to get a list of services.
`SERVER-VERSION`	Prints the version number of the IP$SERVER process.
`STATISTICS`	Prints server usage statistics.
`TIMERS`	Prints debugging information about the various scheduler events in the IP$SERVER process.
`VERBOSE`	Turns on verbose printing of **NETCONTROL** commands and responses.
`VERSION`	Prints the version number of the selected server.

Table 1.4 shows the **NETCONTROL** commands you can use with the ACCESS server.

Table 1.4. NETCONTROL ACCESS Commands

Command	Description
`DEBUG` *`n`*	Sets the ACCESS debugging level. The larger the number, the more verbose the output.
`RELOAD` *`n`*	Rereads your access configuration file `IP$:START_ACCESS.COM` if it is not redefined with the configuration file parameter.

Table 1.5 shows the NETCONTROL ACCOUNTING commands you can use.

Table 1.5. NETCONTROL ACCOUNTING Commands

Command	Description
`ACC-CONTROL-VERSION`	Shows the version of the accounting control logs.
`FILE` *`file_specification>`*	Starts a new accounting control image.
`NOOP`	Does nothing.
`RELOAD`	Restarts the accounting server.
`SHUTDOWN`	Stops the accounting server.

Command	Description
`START`	Starts the accounting server.
`VERSION`	Displays the version of the accounting server control image.

Table 1.6 shows the NETCONTROL commands you can use with the BOOTP server.

Table 1.6. NETCONTROL BOOTP Commands

Command	Description
`DEBUG` *`n`*	Sets the debugging log level.
`DUMP`	Dumps the BOOTP database.
`RELOAD`	Reloads the BOOTP database.

Table 1.7 shows the **NETCONTROL** commands you can use with the CLUSTERALIAS server

Table 1.7. NETCONTROL CLUSTERALIAS Commands

Command	Description
`DEBUG` *`n`*	Sets the CLUSTERALIAS debugging level. The larger the number, the more verbose the output.
`RELEASE` *`ip address`*	Releases the system’s lock of the cluster alias address.
`SHOW` *`n`*	Displays the state of the cluster alias.

Table 1.8 shows the **NETCONTROL** commands you can use with the DHCP client.

Table 1.8. NETCONTROL DHCP Client Commands

Command	Description
`DEBUG` *`value`*	Specifies a decimal integer that is a bitmask of debugging levels used to select messages to pass to OPCOM and the debug log file specified in the DEBUG-FILE parameter. The debugging levels are (in decimal): 1 Fatal Errors 3 3 Errors and Warnings 7 7 Informationals 15 15 Debug Messages 31 31 Dump Packets (Formatted) 63 63 Dump Packets (Hex) By default, Fatal Errors and Warnings are logged.
`VERSION`	Prints the version number of the DHCP Client Control component.
`SHUTDOWN`	Causes the DHCP Client to shut down. This command does not delete the service from the Master Server. It is still registered with NETCONTROL.

Command	Description
`START`	Starts a DHCP Client that has been down by the DHCLIENT **SHUTDOWN** command.

Table 1.9 shows the NETCONTROL commands you can use with the DHCP server.

Table 1.9. NETCONTROL DHCP Commands

Command	Description
`DEBUG` *value*	Specifies a decimal integer that is a bitmask of debugging levels used to select messages to pass to OPCOM and the debug log file specified in the DEBUG-FILE parameter. The debugging levels are (in decimal): 1 Fatal Errors 3 Errors and Warnings 7 Informationals 15 15 Debug Messages 31 31 Dump Packets (Formatted) 63 63 Dump Packets (Hex) By default, Fatal Errors, Errors, and Warnings are logged.
`DHCP-CONTROL-VERSION`	Prints the version number of the DHCP Control component.
`DUMP` *file*	Writes the in-memory configuration of the DHCP server to the file specified or to the DUMPFILE parameter value if *file* is not specified.
`NEWLOG`	Starts a new debug log file, if one is in use.
`PARTNERDOWN`	For Safe-Failover DHCP: Causes the DHCP server to transition into Partner Down state, which indicates that its safe-failover DHCP partner is down.
`RELEASE` *ip-address*	Forces the DHCP server to act as if it heard a DHCP release from the client. This applies to dynamically assigned IP addresses only. Note: the DHCP protocol has no way to tell the client that the address is released, so this command must be used with caution.

`RESTART`
or
`RELOAD`
	Causes the server to restart, at which time it rereads the configuration file.

`SHOW`
`SHOW ALL`
`SHOW CID` *client-identifier*
`SHOW CLIENT` *ip-address*
	Prints a variety of information, depending on the arguments used.
	Takes no arguments. Displays on your computer screen the SHOW SUBNET output for all subnets followed by information about all static assignments in the DHCP server configuration.
	Shows all lease binding and static assignment details for the specified client identifier.

Command	Description
SHOW HADDR *hardware-address* or *-client-identifier* SHOW ISKNOWN SUBCLASS *class-name subclass-data* SHOW SUBNET *ip-address* SHOW LEASES SHOW POOLS	Shows lease binding details for the specified IP address. The IP address must be in a dynamic address pool. Shows all lease binding and static assignment details for the specified hardware address. Shows whether the given hardware address or client identifier is "known", that is if there is a *host* declaration for that hardware address or client address. Shows whether the given subclass data exists as a subclass within the given class. Shows the DHCP address pools for the shared network that *ip-address* is in. Lists each subnet that is on the shared network and each IP address in each pool. Shows all IP addresses with leases (pending, active, or expired) for each shared network. Uses the same format as the **SHOW ALL** command. For each address pool, shows the total number of IP addresses, number of abandoned IP addresses, number reserved for secondary, and number available to be leased. Uses a table format.
SHUTDOWN	Stops the server. This command does not delete the service from the master server. It is still registered with **NETCONTROL**.
START	Starts a server that has been shut down by the DHCP **SHUTDOWN** command.
STATISTICS	This command is supplied only for backward compatibility with previous versions of VSI TCP/IP. It has been superseded by the **SHOW POOLS** command.
UPDATE [(*file*)]	Causes the server to execute the commands in the specified file, if any, or the file specified in the *UPDATEFILE* parameter, if any, or the default file IP$:DHCPD.UPDATES.

Table 1.10 shows the **NETCONTROL** commands you can use with the DOMAINNAME server.

Table 1.10. NETCONTROL DOMAINNAME Commands

Command	Description
DEBUG *n*	Sets the debug level of the Domain Nameserver (the default is no debugging). The larger the number, the more verbose the output. A value of 0 turns off debugging.
DUMP	Dumps the Domain Nameserver cache to the file IP$:DOMAIN-NAME-SERVICE.DB. Use to diagnose database problems.
MAXIMUM-TTL *ttl*	Changes the maximum time-to-live (TTL) that resource records are cached from the default of 604800 seconds (1 week) to the specified value.

Command	Description
`MINIMUM-TTL` *`ttl`*	Changes the minimum time-to-live (TTL) that resource records are cached from the default of zero (0) seconds to the specified value. It is recommended you use this command only if there is a specific need. This could cause problems in that you may be caching resource records for longer than the authoritative administrator intended.
`QUERYLOG`	Toggles query logging ON and OFF. Query logging shows an informational message every time a query is received by the server. Query logging can be directed to OPCOM or a file in the `IP$:NAMED.CONF` file using the logging category *`queries`*.
`RELOAD`	Causes the Domain Nameserver to re-read the configuration file, and subsequently re-read any zone files that have changed.
`RESTART`	Instructs the nameserver to shutdown if it exists, then instructs the master server to start a new nameserver process.
`REWRITE-TTL` *`ttl`*	Sets the time-to-live (TTL) that load balanced resource records are cached from the default of 300 seconds (5 minutes) to the specified value.
`SHOW`	Shows the nodename, address, and rating of any cluster server names.
`START`	Instructs the master server to start the nameserver process.
`STATISTICS`	Appends Domain Nameserver server statistics to the file `IP$:DOMAIN-NAME-SERVICE.STATS` and memory statistics to the file `IP$:DOMAIN-NAME-SERVICE.MEMSTATS`.

`STOP`
or
`SHUTDOWN`
	Stops the server.
	This command does not delete the service from the master server. It is still registered with NETCONTROL.

`VERSION`	Prints the current DNS server version number. (This is the version of BIND from which the VSI TCP/IP DNS server is derived.)

Table 1.11 shows the **NETCONTROL** commands you can use with the EKLOGIN server:

Table 1.11. NETCONTROL EKLOGIN Commands

Command	Description
`DEBUG` *`n`*	Sets the debugging log level.
`FLUSH-CACHE`	Flushes the “KR” services authentication cache.

Table 1.12 shows the **NETCONTROL** commands you can use with the FONTSERVER server.

Table 1.12. NETCONTROL FONTSERVER Commands

Command	Description
`DEBUG`	Sets the FONTSERVER debugging level. The larger the number, the more verbose the output.
`FLUSH`	Removes the fonts loaded into the font server’s cache.

Command	Description
FS-CONTROL-VERSION	Prints the version number of the FONTSERVER control component.
RELOAD	Reloads the font server configuration file IP $:FONT_SERVER.CONFIGURATION.
RESET	Resets the server and closes down all connections to the client.
RESTART	Restarts the server.
START	Starts the server.
SHUTDOWN	Stops the server.

Table 1.13 shows the **NETCONTROL** commands you can use with the IPXRIP server .

Table 1.13. NETCONTROL IPXRIP Commands

Command	Description
ADVERTISE true | false	Advertises non-local routes. The default is “true”.
DEBUG *n*	Sets the debugging log level.
FLUSH	Flushes the non-local routes and updates the interface configuration.
SEND	Sends the IPX RIP packets.
SHOW	Displays the IPX RIP routing table.

Table 1.14 shows the **NETCONTROL** commands you can use with the KLOGIN server.

Table 1.14. NETCONTROL KLOGIN Commands

Command	Description
DEBUG *n*	Sets the debugging log level.
FLUSH-CACHE	Flushes the “KR” services authentication cache.

Table 1.15 shows the **NETCONTROL** commands you can use with the KSHELL server.

Table 1.15. NETCONTROL KSHELL Commands

Command	Description
DEBUG *n*	Sets the debugging log level.
FLUSH-CACHE	Flushes the “KR” services authentication cache.

Table 1.16 shows the **NETCONTROL** commands you can use with the **NETCONTROL** server.

Table 1.16. NETCONTROL NETCONTROL Commands

Command	Description
ACCOUNTING *n*	Disables accounting with an integer value of 0. A positive value enables accounting. The accounting value is checked whenever the master_server starts up a service. The accounting provided is PID, host name, node name, and service name.

Command	Description
`DEBUG` *`n`*	Sets the **NETCONTROL** debugging level. The larger the number, the more verbose the output.

`SHUTDOWN` *`all`*
or
`service name`
	Shuts down all services or shuts down a specific services.

Table 1.17 shows the **NETCONTROL** commands you can use with the NFS v3 Server.

Table 1.17. NETCONTROL NFSV3 Commands

Command	Description
`RESTART`	Restarts the NFS v3 Server process.
`SHUTDOWN`	Stops the server.
`START`	Starts the server.

Table 1.18 shows the **NETCONTROL** commands you can use with the NOT server.

Table 1.18. NETCONTROL NOT Commands

Command	Description
`DEBUG`	Sets the NOT debugging level. The larger the number, the more verbose the output.
`RELOAD`	Reloads the `NOT.CONFIGURATION` or the configuration file the parameter for the NOT service is set to.

Table 1.19 shows the **NETCONTROL** commands you can use with the NTP server.

Table 1.19. NETCONTROL NTP Commands

Command	Description
`DEBUG` *`n`*	Sets the debug level of the NTP server. The larger the number, the more verbose the output.
`NOOP`	Does nothing but verify that the server is running.
`NTP-CONTROL-VERSION`	Displays the version information for the NTP **NETCONTROL** interface.
`PANIC` *`seconds`*	Sets the largest value (in seconds) that will be corrected. The default is 4000 seconds, or just over 66 minutes.
`RELOAD`	Restarts the NTPD server process. Equivalent to SHUTDOWN followed by START.
`SHOW`	Shows the current state of all server and peer connections. Also displays the current value of WAYTOOBIG/PANIC.
`SHUTDOWN`	Causes the NTPD server process to shut down and exit.
`START`	Starts the NTPD server process.
`VERSION`	Displays the version of the NTP server in use.
`WAYTOOBIG`	A synonym for PANIC. Retained for historical reasons.

Table 1.20 shows the **NETCONTROL** command you can use with the PCNFSD server.

Table 1.20. NETCONTROL PCNFSD Command

Command	Description
`DEBUG` *n*	Sets the debugging log level.

Table 1.21 shows the **NETCONTROL** commands you can use with the RACOON server.

Table 1.21. NETCONTROL RACOON Commands

Command	Description
`DEBUG` *n*	Sets the debugging level
`DELETE`	Delete an establish key exchange session.
`ESTABLISH`	Initiate key exchange protocol communication between the remote ip-address and the local ip-address. If local-ip-address is not specified then the value of `IP$HOST_NAME` is used. This does not install Security Associations, but does the initial negotiation necessary to allow Security Associations to be established when necessary. It is not necessary to manually establish the negotiation information – RACOON will do it automatically when necessary.
`FLUSH`	Flush existing key exchange sessions.
`NOOP`	No operation
`SHOW`	Shows the current state of key negotiation between IP addresses
`SHUTDOWN`	Shutdown Racoon
`START`	Start Racoon
`STOP`	Stop Racoon (equivalent to **SHUTDOWN**)
`VERSION`	Version of the control interfac

Table 1.22 shows the **NETCONTROL** commands you can use with the RARP server.

Table 1.22. NETCONTROL RARP Commands

Command	Description
`DEBUG` *n*	Sets the RARP debugging level. The larger the number, the more verbose the output.
`RELOAD`	Reloads the RARP database.

Table 1.23 shows the **NETCONTROL** commands you can use with **NETCONTROL REXEC**, **NETCONTROL RLOGIN**, or **NETCONTROL RSHELL**.

Table 1.23. NETCONTROL "R" Server Commands

Command	Description
`DEBUG`	Sets the "R" SERVICES debugging level. The larger the number, the more verbose the output.
`FLUSH-CACHE`	Flushes the "R" services authentication caches.
`SHOW-CACHE`	Prints the contents of the "R" services authentication caches.

Table 1.24 shows the **NETCONTROL** commands you can use with the RPCBOOTPARAMS server.

Table 1.24. NETCONTROL RPCBOOTPARAMS Commands

Command	Description
`DEBUG` *n*	Sets the debugging log level.
`RELOAD`	Reloads the RPC boot parameters for diskless hosts.

Table 1.25 shows the **NETCONTROL** commands you can use with the RPCLOCKMGR server.

Table 1.25. NETCONTROL RPCLOCKMGR Commands

Command	Description
`DEBUG` *n*	Sets the debugging log level.
`NOOP`	Does nothing; provided for testing the **NETCONTROL** server.
`RPCLOCKMGR-CONTROL-VERSION` *n*	Displays the version number of the RPC Lock Manager Control Server.
`SHOW` *n*	Shows the locks associated with this server.
`START`	Starts the RPC Lock Manager.
`STOP`	Stops the RPC Lock Manager.
`TIMERS` *n*	Displays the timers.

Table 1.26 shows the **NETCONTROL** commands you can use with the RPCMOUNT server.

Table 1.26. NETCONTROL RPCMOUNT Commands

Command	Description
`CLEAR`	Clears the database of clients that have file systems mounted.
`DEBUG` *n*	Sets the RPCMOUNT debugging level. The larger the number, the more verbose the output.
`DUMP`	Forces RPCMOUNT to write the current mount database to the on-disk cache; it is normally only written every few minutes.
`RELOAD`	Reloads export and restriction databases from the `NFS_EXPORT.DAT, NFS_GROUP.DAT,NFS_MNTLST.DAT,` and `NFS_PROXY.DAT` files.
`SHOW`	Prints the current mount database.

Table 1.27 shows the **NETCONTROL** commands you can use with the RPC Portmapper server.

Table 1.27. NETCONTROL RPCPORTMAP Commands

Command	Description
`DEBUG` *n*	Sets the RPCPORTMAP debugging level. The larger the number, the more verbose the output.
`SHOW`	Prints the current portmap database.

Table 1.28 shows the **NETCONTROL** command you can use with the RPCQUOTAD server.

Table 1.28. NETCONTROL RPCQUOTAD Command

Command	Description
`DEBUG` *`n`*	Sets the debugging log level.

Table 1.29 shows the **NETCONTROL** commands you can use with the RPCSTATUS server.

Table 1.29. NETCONTROL RPCSTATUS Commands

Command	Description
`DEBUG` *`n`*	Sets the RPCSTATUS debugging level. The larger the number, the more verbose the output.
`RELOAD`	Reloads the RPCSTATUS database.
`SHOW`	Shows the parameters governing RPCSTATUS.
`SIMULATE-CRASH`	Causes the server to notify all monitoring clients that the system has crashed. Do not use this command without first contacting VSI Technical Support.

Table 1.30 shows the **NETCONTROL** commands you can use with the SNMP server.

Table 1.30. NETCONTROL SNMP Commands

Command	Description
`RELOAD`	Causes the SNMP Agent to reread the configuration file.
`SHUTDOWN`	Stops the SNMP Agent. This command does not delete the service from the master server. It is still registered with **NETCONTROL**.
`SNMP-CONTROL-VERSION`	Prints the version number of the SNMP Agent component.
`START`	Starts a SNMP Agent that has been shut down by the **SNMP SHUTDOWN** command.

Table 1.31 shows the **NETCONTROL** commands you can use with the SSH server.

Table 1.31. NETCONTROL SSH Commands

Command	Description
`DEBUG`	Toggles debugging on/off in all SSHD daemon processes running on the server.
`MASTER_RESTART`	Stops and restarts only the SSHD Master process. All other SSH processes and users are not affected.
`RESTART`	Stops and restarts the server. This stops not only the SSHD_MASTER process but also all SSHD processes running on the server, which has the effect of logging out all SSH sessions currently active on the server.

`SHOW`	Displays information on all executing daemon processes. For example:
	`RAPTOR_$ ip netcontrol ssh show`
	`Connected to NETCONTROL server on "LOCALHOST"`

Command	Description
	< raptor.psccos.com Network Control V10.5(10) at Tue 09-May-2017 11:41AM-MDT < < SSHD Master PID = 20800099 < < Process "SSHD 0000" (pid 20800156) < User = OREILLY < From system 217.225.329.75 port 1064 < Started: 05/09/2002 17:41:43 < Bytes in: 23 out: 425 (from child process: 425) < PTD Device = _FTA5: < Current child process = "OREILLY_@FTA5" (pid 20800157) < < End of Show SSH RAPTOR_$
SHUTDOWN	Stops the server. This stops not only the SSHD_MASTER process but also all SSHD processes running on the server, which has the effect of logging out all SSH sessions currently active on the server.
START	Starts the server.

Table 1.32 shows the **NETCONTROL** commands you can use with the SYSLOG server.

Table 1.32. NETCONTROL SYSLOG Commands

Command	Description
DEBUG *n*	Sets the debugging log level.
RELOAD	Reloads the configuration file.

Table 1.33 shows the **NETCONTROL** commands you can use with the TELNET server.

Table 1.33. NETCONTROL TELNET Commands

Command	Description
DEBUG	Sets the TELNET debugging level. The larger the number, the more verbose the output.
TRACE *n*	Sets the trace level.

Table 1.34 shows the **NETCONTROL** commands you can use with the TFTP server.

Table 1.34. NETCONTROL TFTP Commands

Command	Description
DEBUG *n*	Sets the TFTP debugging level. The larger the number, the more verbose the output.
RELOAD	Reloads the TFTP server parameters (the default TFTP directory and `IP$:TFTP.FILENAME-TRANSLATIONS`).
SHOW	Prints the current status of the TFTP server.

Command	Description
`SHOW-TRANSLATION`	Shows the TFTP file name translation table.

Table 1.35 shows the **NETCONTROL** command you can use with the UCXQIO server.

Table 1.35. NETCONTROL UCXQIO Command

Command	Description
`DEBUG` *`n`*	Sets the UCXQIO debugging level. The larger the number, the more verbose the output.

Table 1.36 shows the **NETCONTROL** commands you can use with the XDM server.

Table 1.36. NETCONTROL XDM Commands

Command	Description
`DEBUG`	Sets the XDM debugging level. The larger the number, the more verbose the output.
`RELOAD`	Causes the XDM server to reload its configuration file.
`RESTART`	Restarts the XDM server.
`SHOW`	Shows the status of all managed displays.
`START`	Starts the server.
`SHUTDOWN`	Stops the server.
`XDM-CONTROL-VERSION`	Prints the version number of the XDM Control component.

IP NFSDISMOUNT

IP NFSDISMOUNT — Dismounts a remotely mounted NFS file system.

Format

`IP NFSDISMOUNT` *`mount_device`*

Parameter

`mount_device`

Specifies an NFS *`x`*: device associated with a remotely mounted file system.

Qualifiers

`/ALL` [*`host_name`*]

Specifies that **NFSDISMOUNT** notifies remote systems that no file systems are currently mounted (this is usually used as part of the reboot procedure).

NFSDISMOUNT /ALL does not dismount file systems, but rather notifies an NFS server that the local system does not have any mounted. If you do not specify host_name, **TRACEROUTE** broadcasts the request to the local network.

`/LOG`

Specifies that **NFSDISMOUNT** displays information when a dismount occurs.

Example

This example shows how to dismount a remotely mounted file system attached to the local mount device NFS3:.

```
$ IP NFSDISMOUNT NFS3:
```

IP NFSMOUNT

IP NFSMOUNT — Mounts a remote NFS file system so it can be used locally. NFSMOUNT requires CMKRNL, SETPRV, SYSPRV, SYSNAM, ALTPRI, DETACH, ACNT, and SYSLCK privileges.

Format

`IP NFSMOUNT` *`node mount_point logical_name`*

Parameters

`node`

Specifies the name of the computer serving the file system to the network.

`mount_point`

Specifies the portion of an NFS file system to be mounted. The format of the specified mount point depends on the server. Enclose mount_point in quotes if it contains special or lowercase characters.

`logical_name`

Specifies an OpenVMS logical name to assign to the mount device. OpenVMS users can access remote files using this logical name.

Qualifiers

`/FID_CACHE=`*`size`*

Specifies the size (in bytes) of the File Identifier (FID) cache. Values range from 5 to 5000000. The default is 10000.

`/LOCKING=( [`*`local,`*`] [`*`network`*`] )`

Specifies the type of file locking used on the NFS-mounted file system.

- *`LOCAL`* — Specifies that OpenVMS file-locking operations will only be consistent on the local system.
- *`NETWORK`* — Uses the NFS Network Lock Manager to obtain consistent file locking between all NFS clients. Use of the **/LOCKING=NETWORK** qualifier adds protocol overhead.

If you specify both `LOCAL` and `NETWORK`, network locking is attempted; if the remote lock manager cannot be contacted, local locking is used.

/PAGEFILE=*pages*

Specifies the page file quota for the NFS_CLIENT_ACP process. This process is created when the first NFS file system is mounted. This qualifier is ignored on subsequent mounts. The default is 65535 pages.

/PORT=*port_number*

Specifies the remote port to connect to for NFS service.

/PRIORITY=*priority*

Specifies the base process priority of the NFS_CLIENT_ACP process created when the first NFS file system is mounted. The qualifier is ignored on subsequent mounts. The default is 7.

/PROCESSOR={UNIQUE | SAME}

Determines whether a separate ACP process is created for every NFS device. This mechanism allows NFS devices to function in parallel so that one NFS device does not have to wait for an NFS operation on another NFS device to complete. Multiple ACPs allow for multiple outstanding I/O, and operations happen in parallel.

A setting of UNIQUE creates a separate NFS_CLIENT_*n* process for each mount, where *n* is the number of the NFS device (such as NFS_CLIENT_2, which corresponds with the device NFS2).

A setting of **/PROCESSOR=SAME=*nfs_device*** assigns the mount to the same ACP process as the specified *nfs_device*. For example, **/PROCESSOR=SAME=NFS3** assigns this mount to the NFS_CLIENT_3 ACP process.

Any mounts specified without the **/PROCESSOR** qualifier use a single, default process.

It is recommended that you use the **/PROCESSOR** qualifier to group mounts based on the remote server. That way, if the server goes down, it does not cause access to other servers to hang. (You can use the **/SOFT** qualifier to permit NFS operations to time out instead of hanging indefinitely.)

/READ_SIZE=*read_size*

Specifies the maximum size of the read operations the NFS Client performs. The default, 8192, is correct for most servers.

/RELOAD

Instructs the NFS client software to reload its UID translation table from the NFS configuration file.

/SEMANTICS=(

[*advisory_close*]

[*case_insensitive_filenames*]

[*nofdl_files*]

[*nolinks*]

[*nostream_conversion*]

[*nounique_fileno*]

[*noversions*]

[*novms_access_checking*]

[*preserve_dates*]

[*upper_case_default*]

[*vms_filenames*]

[*vms_server*]

Specifies the capabilities and characteristics of the NFS Server that control the behavior of the VSI TCP/IP NFS Client, as described in the following table.

Attribute	Description
ADVISORY_CLOSE	Sends a OpenVMS server a command to close the file when there are no more references to it on the client.
CASE_INSENSITIVE_FILENAMES	Specifies that UNIX files accessed by an OpenVMS system not have their file names converted using the conversion characters (see HELP IP File_Name_Character_Map for a list of these characters). Use this option when an NFS server treats all file names as case-insensitive. When this option is set, all file names accessed through NFS are converted to lowercase. When returned to the server, they are handled in lowercase. The NFS server must be able to accept lowercase file names. This option is disabled by default.
NOFDL_FILES	Disables the use of ".fdl" files by the VSI TCP/IP NFS Client to store RMS attributes. This option must be used if the NFS server doesn't allow these file names. Its use severely limits the ability of the NFS Client to store record attributes.
NOLINKS	Disables the automatic creation of hard links to the latest version of a file. The NFS Client normally uses a hard link operation to link the top version of a file name "foo.bar;12" to the unversioned name "foo.bar" for more convenient access from the NFS Server side. This option may be used either to reduce the overhead in creating it or if the NFS Server does not support hard links.
NOSTREAM_CONVERSION	Disables the automatic conversion of text files to STREAM format. The NFS Client normally converts requests to create Variable Length Record Carriage Return Carriage Control files into requests to create Stream files. This option disables this conversion.
NOUNIQUE_FILENO	Specifies whether or not the NFS Server is to generate unique file numbers for each file (most NFS servers do). If the client knows that file numbers are unique, it uses a faster algorithm to refresh stale directory entries in the cache. Use

Attribute	Description
	of this qualifier disables the faster refresh algorithm, and is equivalent to the **/NOUNIQUE_FILENO** qualifier.
NOVERSIONS	Disables support for multiple file versions. The NFS Client normally stores multiple versions of OpenVMS files by using the semicolon character in the file name on the NFS Server side. You must use this option to disable the ability to create multiple versions of files if the NFS Server does not support file names with the semicolon character.
NOVMS_ACCESS_CHECKING	Specifies that the client does not perform a full OpenVMS access check, including a check for ACLs and security alarms. If this option is not specified, the NFS Client considers ACLs and security alarms when granting or denying access.
PRESERVE_DATES	Allows you to store OpenVMS-style dates and times for files.
UPPER_CASE_DEFAULT	Assumes file names are in uppercase on the server until it sees the $ character used to toggle case.
VMS_FILENAMES	Specifies that the NFS Client should not perform the usual mapping between OpenVMS and UNIX-style file names. This option can be used to permit all OpenVMS file names to be stored using the NFS client; however, its use prevents the NFS Client from being used to access files which do not conform to the OpenVMS file name conventions.
VMS_SERVER	Specifies that the NFS server is a VSI TCP/IP NFS Server of revision V3.0 or later and supports OpenVMS-specific extensions to the NFS protocol to store file attributes. If the NFS Server does not support these extensions, the mount will fail. This option is equivalent to the **/VMS_SERVER** qualifier and overrides any other semantics specified.

`/SOFT`

Specifies that, if the NFS client is unable to reach the NFS server after the time period specified by /**TIMEOUT**, an error is returned to the user (SS$_UNREACHABLE). If the file system is mounted without the /**SOFT** qualifier, the NFS client retries the operation forever.

`/TIMEOUT=`*`timeout`*

Specifies the total time, in tenths of a second, that it takes for an RPC request to timeout. Retries are attempted via UDP for an interval of one-fifth the value specified for /**TIMEOUT**. The minimum value allowed for this setting is 30 tenths of a second. /**TIMEOUT** does not affect TCP timeouts.

`/TRANSPORT=` [*`tcp`*] [*`udp`*]

Specifies the underlying transport used for the NFS requests. (The default is UDP if /**TRANSPORT** is not specified.) The TCP transport can be used with servers that support it. If you specify both transports, TCP is tried first; if it fails, the mount uses UDP.

`/UNIQUE_FILENO (default)`

`/NOUNIQUE_FILENO`

Specifies whether or not the NFS Server is to generate unique file numbers for each file (most NFS servers do). If the NFS Client knows that file numbers are unique, it uses a faster algorithm to refresh stale directory entries in the cache. The **/NOUNIQUE_FILENO** qualifier is equivalent to /**SEMANTICS=NOUNIQUE_FILENO**.

`/VMS_SERVER`

Specifies that the NFS server is a VSI TCP/IP NFS Server of revision V3.0 or later and supports OpenVMS-specific extensions to the NFS protocol to store file attributes. If the NFS Server does not support these extensions, the mount fails. This qualifier is equivalent to /**SEMANTICS=VMS_SERVER** and overrides any other semantics specified.

`/VOLUME=`*`volume_name`*

Specifies the display name of the mounted volume (which appears via **SHOW DEVICE**). (The default is the remote mount_point name.)

`/WRITE (default)`

`/NOWRITE`

Specifies whether or not the file system is to be mounted for both read and write access. /**NOWRITE** prevents users from modifying the file system.

`/WRITE_SIZE=`*`write_size`*

Specifies the maximum size of packets written by the NFS client. The default, 8192, is correct for most servers.

`/WSEXTENT=`*`pages`*

Specifies the working set extent for the `NFS_CLIENT_ ACP` process. This process is created when the first NFS file system is mounted. The qualifier is ignored on subsequent mounts. The default is 20000 pages.

`/WSQUOTA=`*`pages`*

Specifies the working set quota for the NFS_CLIENT_ACP process. This process is created when the first NFS file system is mounted. The qualifier is ignored on subsequent mounts. The default is 2000 pages.

Examples

This example shows how to mount the remote file system "/usr" on the server named "sunset" on the local mount device NFS3:.

```
$ IP NFSMOUNT SUNSET "/usr" disk$sunset
%NFSMOUNT-I-MOUNTED, /usr NFS mounted on _NFS3:[000000]
$
```

This example illustrates the use of /**PROCESSOR=UNIQUE**, creating four ACP processes-one for each device.

```
$ IP NFSMOUNT/VMS/PROCESSOR=UNIQUE SCOOBY USERS: SCOOBY1
$ IP NFSMOUNT/VMS/PROCESSOR=UNIQUE SCOOBY USERS2: SCOOBY2
$ IP NFSMOUNT/VMS/PROCESSOR=UNIQUE SHAGGY USERS:  SHAGGY1
$ IP NFSMOUNT/VMS/PROCESSOR=UNIQUE SHAGGY USERS2: SHAGGY2
```

This example illustrates the use of **/PROCESSOR=SAME**. In this example, all access to the server named SCOOBY goes through one ACP process, and all access to SHAGGY goes through another process.

```
$ IP NFSMOUNT/VMS SCOOBY USERS: SCOOBY1
$ IP NFSMOUNT/VMS/PROCESSOR=SAME=SCOOBY1 SCOOBY USERS2: SCOOBY2
$ IP NFSMOUNT/VMS SHAGGY USERS: SHAGGY1
$ IP NFSMOUNT/VMS/PROCESSOR=SAME=SHAGGY1 SHAGGY USERS2: SHAGGY2
```

IP NSLOOKUP

IP NSLOOKUP — Performs test queries on the domain name service (DNS) system.

Format

`IP NSLOOKUP` [*name*] [*nameserver*]

Description

When invoked with no parameters, **IP NSLOOKUP** allows commands to be run interactively. Table 1.37 lists the commands that can be run in interactive mode

Table 1.37. NSLOOKUP Commands

Command	**Description**
name	Prints information about *name* using the default server.
name server	Prints information about *name* using *server*.
exit	Exits NSLOOKUP.
finger [*user*]	Finger the optional *user* at the current default host.
or**?**	Prints help information.
set all	Prints the current status of all options.
set class=*class*	Sets the query class to one of these: IN, CHAOS, HESIOD, or ANY.
set [no]debug	Prints debugging information.
set [no]d2	Prints exhaustive debugging information.
set [no]defname	Appends the domain name to each query.
set [no]recurse	Asks for a recursive answer to a query.
set [no]vc	Always uses a virtual circuit.
set domain=*name*	Sets the default domain name to *name*.
set port=*port*	Sets the port number on which to send a query.
set root=*name*	Sets the root name server to *name*.
set retry=*n*	Sets the number of retries to *n*.
set srchlist=*name1*[/*name2*/...*name6*]	Sets the domain to *name1* and the search list to *name1* through *name6*.

Command	Description
`set timeout=`*`n`*	Sets the timeout interval to *`n`*.
set query-type=*type* or **set type=*type***	Sets the resource record (RR) type to query for. See Table 1-35.
`server `*`name`*	Sets the default server to *`name`*, using the current default server.
`lserver name`	Sets the default server to *`name`*, using the original default server.
`root`	Sets the current default server to the root.
`ls [`*`option`*`] name [>`*`file`*`]`	Lists the domain *`name`*, with output optionally going to *`file`*. *`option`* is one of the following: • a List fully-qualified 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 (such as A, CNAME, and MX)

Parameters

[*name*]

Specifies a host or domain name.

[*nameserver*]

Specifies the name server to query.

Qualifiers

/CLASS=*recordclass*

Specifies which CLASS records are asked for. Valid classes are ANY, IN, CHAOS, and HESIOD. (The default is **/CLASS=IN**, Internet records.)

/DEBUG

/NODEBUG (default)

Causes the resolver to print debugging information, including formatted responses.

/DEBUG2

/NODEBUG2 (default)

Causes the resolver to print formatted queries, and additional, less useful debugging information.

/DEFNAMES (default)

`/NODEFNAMES`

Specifies that the resolver adds this system's domain name to any name not explicitly terminated with a period. **/DEFNAMES** is the default.

`/DNSRCH (default)`

`/NODNSRCH`

Specifies that the resolver searches up the domain tree from this system's name for any name not explicitly terminated with a period.

`/DOMAIN=`*`domainname`*

Specifies a default domain other than the domain of this host.

`/IGNTC`

`/NOIGNTC (default)`

Tells the resolver to ignore truncation in responses.

`/PORT=port`

Specifies a port other than the standard nameserver port of 53.

`/RECURSE (default)`

`/NORECURSE`

Requests that the name server use recursion to answer the query.

`/RETRY=`*`retrycount`*

Specifies the number of retries the resolver makes when querying a name server via UDP (by default, 4).

`/ROOT_SERVER=`*`rootservername`*

Specifies a root name server other than `A.ROOT-SERVERS.NET`.

`/TIMEOUT=`*`seconds`*

Specifies a different period to wait for responses. The default is 4 seconds.

`/TYPE=`*`recordtype`*

Specifies which **TYPE** resource records are asked for. The default is **/TYPE=A** (address records).

All standard DNS record types are supported. Table 1.38 for Ngives a partial list of valid values for the /TYPE qualifier.

Table 1.38. Sample Resource Record Types for NSLOOKUP/TYPE Qualifier

Resource Record	Description	Resource Record	Description
A	Address records	MR	Mail rename domain name

Resource Record	Description	Resource Record	Description
ANY	Any	MX	Mail exchanger
AXFR	Zone transfer	NS	Authoritative name server
CNAME	Fully-qualified name for an alias	PTR	Domain name pointer
GID	Group ID	SOA	Start of a zone of authority
HINFO	Host information	TXT	Arbitrary text
MAILB	Mailbox for a user	UID	User ID
MB	Mailbox domain name	UINFO	Arbitrary user information
MG	Mail group member	WKS	Well-known service description
MINFO	Mailbox or mail list information		

`/VC`

`/NOVC (default)`

Specifies that the resolver uses virtual circuits instead of datagram queries.

IP NSUPDATE

IP NSUPDATE — Performs dynamic updates to the domain name service (DNS) server.

Format

`IP NSUPDATE` [*`filename`*]

Description

NSUPDATE can read commands from a specified file or from the terminal.

NSUPDATE can be used with the UNIX-style syntax by defining it as a foreign command:

```
$ NSUPDATE :== $IP$:NSUPDATE
```

Both the UNIX-style options and the OpenVMS qualifiers are listed below.

NSUPDATE reads input records, one per line, each line contributing a resource record to an update request. All domain names used in a single update request must belong to the same DNS zone. A blank line causes the accumulated records to be formatted into a single update request and transmitted to the zone's authoritative name servers. Additional records may follow, which are formed into additional, completely independent, update requests. For the last request to be transmitted, a blank line must end the input.

Records take one of two general forms:

- *Prerequisite* records specify conditions that must be satisfied before the request will be processed.
- *Update* records specify changes to be made to the DNS database.

An update request consists of zero or more prerequisites and one or more updates. Each update request is processed atomically, that is, all prerequisites must be satisfied before all updates will be performed.

NSUPDATE understands the input record formats listed in the following table.

Table 1.39. NSUPDATE Commands

Command	Description
`prereq nxdomain` *`name`*	Requires that no RR of any type exists with name *`name`*.
`prereq nxrrset` *`name`* [*`class type`*]	Requires that no RR exists of the specified *`type`* and *`name`*.
`prereq yxdomain` *`name`*	Requires that at least one RR named *`name`* must exist.
`prereq yxrrset` *`name`* [*`class`*] *`type`* [*`data`*]	Requires that a RR exists of the specified *`type`* and *`name`*. If *`data`* is specified, it must match exactly.
`update add` *`name ttl`* [*`class`*] *`type data`*	Adds a new RR with specified *`ttl`*, *`type`*, and *`data`*.
`update delete` *`name`* [*`class`*] [*`type`*] [*`data`*]	Deletes RRs named *`name`*. If *`type`* (and possibly *`data)`* is specified, only matching records will be deleted.

Parameters

[*`filename`*]

Specifies a file containing NSUPDATE commands to be executed.

Qualifiers

`-d`

`/DEBUG`

`/NODEBUG (default)`

Causes the resolver to print debugging information.

`-k` *`keydir+keyname`*

`/KEY=(KEYNAME=`*`key`* [KEYDIR=*`directory`*]])

Specifies a TSIG key for NSUPDATE to use to sign its updates. The default value for KEYDIR is the current default directory.

Note

On Unix, the syntax is *`keydir:keyname`*. On OpenVMS, the colon is replaced by a plus sign (+). The *`keyname`* must be specified to match the key and private filenames, with periods instead of dollar signs. This may not match the domainname if DNSKEYGEN had to abbreviate it to fit into an OpenVMS file name.

`-v`

`/VC`

`/NOVC (default)`

Specifies that the resolver uses virtual circuits (TCP) instead of datagram (UDP) messages.

Examples

The following example illustrates the interactive use of NSUPDATE to change an IP address by deleting any existing A records for a domain name and then inserting a new one. Since no prerequisites are specified, the new record will be added even if there were no existing records to delete.

Note

The trailing blank line is required to process the request.

```
$ IP nsupdate
> update delete test.example.com A
> update add test.example.com 3600 A 10.1.1.1
>
```

In this example, a CNAME alias is added to the database only if there are no existing A or CNAME records for the domain name.

```
$ IP nsupdate
> prereq nxrrset www.example.com A
> prereq nxrrset www.example.com CNAME
> update add www.example.com 3600 CNAME test.example.com
>
```

IP PING

IP PING — Sends ICMP Echo Request packets to the specified host to measure network packet loss and latency.

Format

`IP PING` *`host`*

Description

IP PING returns the following status codes:

Status Code	Description
SS$_NORMAL	Successful PING.

Status Code	Description
SS$_IVBUFLEN	An invalid length was specified on the **/DATA_LENGTH** qualifier. The maximum value is 65468.
SS$_NOSUCHNODE	Failed attempt to PING an unknown host.
SS$_PROTOCOL	Remote system is not configured to support ICMP.
SS$_NOPRIV	Access to PING denied by the system manager.
SS$_DATALOST	Some PING responses were received, but some were lost; that is, a PING success rate of less than 100%.
SS$_UNREACHABLE	No responses were received.

IP PING6 performs the same function for IPv6 networks, sending ICMP6 packets over IPv6.

Parameter

host

Specifies the host to ping.

Qualifiers

/ADDRESSES

Sends a node information query packet instead of an ICMP6 Echo Request to request the addresses which the host responds to. Not all systems support node information query packets. This qualifier is only valid for **IP PING6**.

/DATA_LENGTH=*number-of-bytes*

Specifies the number of bytes of data to attach to ICMP Echo Request packets. If not specified, a reasonable default value is supplied. Increase the **DATA LENGTH** to check for gateways that do not fragment IP packets correctly.

/DEBUG

/NODEBUG (default)

Enables socket-level debugging in the VSI TCP/IP kernel. This qualifier is usually only useful for debugging the VSI TCP/IP kernel.

/IPV6

Specifies that an IPv6 ping is desired. (The default is IPv4.)

/FLOOD

Indicates that **IP PING** is used to flood the network with ICMP Echo packets. **IP PING /FLOOD** transmits these packets 100 times per second or whenever a response is received. Requires SYSPRV privilege.

/NUMBER_OF_PACKETS=*number_of_packets_to_send*

Specifies the number of ICMP Echo Responses received before terminating. If not specified, IP PING runs until you press **Ctrl**/**C**.

/PRELOAD=*number_of_packets_to_send*

Specifies the number of packets sent in rapid succession before entering the normal mode of operation.

/QUIET

/NOQUIET (default)

Causes **IP PING** to not display information when packets are received.

/RECORD_ROUTE

Displays a list of IP routers that the ICMP Echo Request packets traverse. This qualifier uses the IP record route option to display a list of IP routers that the ICMP Echo Request packet traverses. Not all implementations of IP handle this option correctly, so the use of **/RECORD_ROUTE** may result in a garbled response.

Note

The record route IP option is not supported correctly by 4.3 BSD-derived Internet hosts, including VSI TCP/IP prior to Version 5.0. Use the **IP TRACEROUTE** utility to find the path between two hosts.

/ROUTE (default)

/NOROUTE

Disables IP routing of ICMP packets. The default, /ROUTE, allows IP routing to get the packet to destinations separated by gateways.

/VERBOSE

/NOVERBOSE (default)

Displays extra information as ICMP packets are sent or received.

Examples

This example shows using PING to test the round-trip delay to a distant host.

```
$ IP PING TRUTH.GREEN.AC.N
PING TRUTH.GREEN.AC.NZ (130.217.64.3) : 56 data bytes
64 bytes from 130.217.64.3: icmp_seq=1 time=670 ms
64 bytes from 130.217.64.3: icmp_seq=2 time=670 ms
64 bytes from 130.217.64.3: icmp_seq=3 time=670 ms
64 bytes from 130.217.64.3: icmp_seq=4 time=650 ms <Ctrl/C>
----TRUTH.GREEN.AC.NZ PING Statistics----
4 packets transmitted, 4 packets received, 0% packet loss
round trip (ms)  min/avg/max = 650/663/670
```

IP RMTALLOC

IP RMTALLOC — Provides local access to a remote tape or CD-ROM device on the specified remote host. RMTALLOC does not actually read from or write to the magnetic tape, or read from the CD-ROM; other programs supplied with the OpenVMS Operating System provide this support. Optionally, RMTALLOC can associate a logical name with the device.

Format

`IP RMTALLOC` *`host.domain`*`[::][["]["]][`*`logical-name:`*`]`

Parameters

`host.domain`

Specifies the remote host name on which the tape or CD-ROM device is allocated (the domain name-either just the domain name or the fully qualified domain name-of the remote host).

`[["]`*`device-name`*`]`

Specifies an optional device name entered with single or double colons. If the device name contains special characters, such as a UNIX-style device name (/dev/rst8), enclose the name in double quotes ("/dev/rst8").

`[`*`logical-name:`*`]`

Specifies the name associated with the device. Use a name you created or one designated by your system manager. The string is from 1 to 255 alphanumeric characters. If the string contains spaces, enclose the string in single quotes. (Do not use trailing colons.) The logical name you specify becomes a process name, with the device name as the equivalence name. The logical name remains defined until it is explicitly deleted or until your process terminates.

Qualifiers

`/CD`

`/NOCD (default)`

Specifies that the remote device is a CD-ROM rather than a tape device. When /CD is specified, the local device takes the RCD*`xxx`*: name. When **/NOCD** is specified, the local device takes the RMT: name. The remote system is tested to ensure that the specified device type exists; if not, an error displays and RMTALLOC fails.

Note

/CD cannot be used with either the **/SEMANTICS** or **/WRITE** qualifiers.

`/LOG (default)`

`/NOLOG`

Displays a message indicating the name of the local device allocated, and the official host name of the remote host and device name.

`/PASSWORD[=`*`password`*`]`

Specifies the password to use to access the remote host. You may optionally specify the password as the qualifier value (which is not recommended). Specifying **/PASSWORD** without the value causes the password to be prompted for and read without echoing it (if the current input device supports it). If present, this qualifier causes RMTALLOC to use the REXEC server on the remote host rather than the RSHELL server.

`/SEMANTICS=`

[*`blocksize=blocksize`*]

[comment="*`comment`*"]

[density=*`density`*]

[label="*`label`*"]

[*`[no]mount`*]

[*`[no]rewind`*]

[*`[no]unload`*]

Specifies attributes for a magnetic tape device. Do not use with the **/CD** qualifier. Enter keywords separated by commas and enclosed in parentheses. These values pass information to the system operator at the remote system. For example, the values in LABEL and COMMENT display on the remote system console and request that the tape name indicated by LABEL be mounted.

Attribute	**Description**
`BLOCKSIZE`	Specifies the remote tape blocksize.
`COMMENT`	Specified as a string enclosed in double quotes; the information is displayed in the remote OPCOM message, either appended to or replacing the default text, depending on whether the resulting length is less than the maximum of 78 characters. Supplying the COMMENT value is the only way you can send a tape-specific message to the remote operator. The OPCOM message from the DCL **MOUNT/COMMENT** command is not passed to the remote RMT server; this message is only sent to OPCOM for a local operation. The default RMTALLOC command causes the remote tape to be mounted foreign, causing an OPCOM message to be generated if the tape drive is offline. The default RMTALLOC command is equivalent to the RMTALLOC **/SEMANTICS=MOUNT** command, which causes RMTALLOC to not complete until a tape has been physically loaded and the drive is online. Therefore, use the COMMENT value to ensure that the operator is informed of your request. Override the RMTALLOC default with the RMTALLOC **/SEMANTICS=NOMOUNT** command, which allocates the tape unit but does not wait for completion. Without the comment, RMTALLOC provides user, node, and device information, as shown in this example:

Attribute	Description
	%%% OPCOM 25-MAR-2017 11;24:35.46 %%% FROM NODE WHORFIN AT 25-MAR-2017 11:24:35.44 REQUEST 87, FROM USER HOLMES ON WHORFIN Please mount device _WHORFIN$ mka500 : RMT tape service request from WHORFIN.FLOWERS.COM
DENSITY	Specifies the density in bits per inch (BPI).
LABEL	Indicates the name by which the tape is known to the remote system. This could be your name, a site-specific numbering scheme, and so on.
[NO]MOUNT	Indicates whether the tape needs to be mounted. This option does not replace the MOUNT command; it only means that RMTALLOC should continue until the remote tape is mounted.
[NO]REWIND	Indicates whether the tape must be rewound before or after use.
[NO]UNLOAD	Indicates whether the tape must be unloaded from the drive after use.

Note

The remote tape drive must be able to write variable length blocks to permit OpenVMS BACKUP to work correctly. Sun QIC tapes cannot do this and do not work with the OpenVMS BACKUP utility. The **RMTALLLOC /SEMANTICS=NOMOUNT** command does not work correctly with multivolume BACKUP save sets. When using RMTALLOC to allocate a remote OpenVMS TMSCP tape drive, the OpenVMS COPY utility cannot copy files from a tape if the TMSCP tape drive is served from a different node than the one specified in the RMTALLOC command.

/TRUNCATE_USERNAME

/NOTRUNCATE_USERNAME (default)

Truncates OpenVMS user names to eight characters or less. Under the UNIX Operating System, the remote user name has a maximum of eight characters. If a longer user name is supplied to such a system, a "remuser too long" error results and RMTALLOC fails.

/UNIX_SERVER=*value*

Specifies that RMTALLOC provide special handling for systems with problematic tape devices. Accepted values are:

Value	Description
BROKEN	Enables one OpenVMS BACKUP save set to be written to a remote UNIX tape. Use this value for SunOS 4.1 and SunOS 4.1.2. May also be useful on other UNIX-incompatible tapes and servers.
UNIX	Enables full OpenVMS tape functionality on an ULTRIX tape drive.

/USERNAME=*remote-username*

Specifies the remote user name to which you want to log in. If not specified, the default is the user name associated with your process.

`/VMS_ATTRIBUTES (default)`

`/NOVMS_ATTRIBUTES`

Verifies whether the remote RMT server is also running VSI TCP/IP. If it is, RMT uses an improved RMT protocol to transfer OpenVMS device attributes and I/O completion status values between your system and the remote host. Because this negotiation is compatible with UNIX Operating System implementations of RMT (including BSD and SunOS), it is enabled by default, but may be disabled if compatibility problems arise.

`/WRITE (default for mag tapes)`

`/NOWRITE (default for CD-ROMs)`

Specifies that the tape is not write-protected; if **/NOWRITE** is specified, the tape is write-protected. /**WRITE** cannot be specified with **/CD**.

Examples

This example illustrates the use of the OpenVMS TAR utility. (VMS TAR is a public domain program available from CETS.) First the tape is allocated with RMTALLOC, then the drive is mounted. Next, a file is written to the tape, the tape contents are listed, and the file is extracted back from the tape. Finally, the tape is dismounted and deallocated.

```
$ RMTALLOC CONE.FLOWERS.COM::MUA0: MYTAPE
%RMT-I-ALLOC, _MYSYS$RMT1: allocated (CONE.FLOWERS.COM::MUA0:)

$ MOUNT /FOREIGN /RECORD_SIZE=512 /BLOCK_SIZE=10240 MYTAPE
%MOUNT-I-MOUNTED,  MYTAPE mounted on _MYSYS$RMT1:

$ TAR /ARCHIVE=MYTAPE WRITE AFILE.TXT
%TAR-S-WRITTEN, written USERS:[ME]AFILE.TXT;1 (13495 bytes)
%TAR-S-TOTWRITE,  total of 1 file written

$ TAR LIST /ARCHIVE=MYTAPE
Listing of archive _MYSYS$RMT2:
-rw------    0/       0     13495  24 Apr 2017 14:31  afile.txt
Total of 1 files listed, 1 files in archive.

$ TAR /ARCHIVE=MYTAPE EXTRACT AFILE.TXT
%TAR-S-TOTCREAT,  total of 0 files created, 1 file scanned

$ DISMOUNT _MYSYS$RMT1:

$ DEALLOCATE _MYSYS$RMT1:
This example illustrates how to allocate access to a UNIX tape.

$ RMTALLOC FOO::"/deV/rst42" UNIXTAPE
%RMT-I-ALLOC, _MIURA$RMT7: allocated (FOO.BAR.COM::/dev/rst8)
$
```

This example illustrates how to allocate access to a UNIX tape.

```
$ RMTALLOC FOO::"/deV/rst42" UNIXTAPE
%RMT-I-ALLOC, _MIURA$RMT7: allocated (FOO.BAR.COM::/dev/rst8)
$
```

This example allocates remote UNIX operating system tape device /dev/rst42 on host FOO.BAR.COM and associates UNIXTAPE with the _MIURA$RMT7 local pseudo-device.

```
$ RMTALLOC/CD/NOWRITE CONTROL::DISK$CD: -

_$ DISK$CONTROL_CD/USER=SYSTEM
%RMT-I-ALLOC _GRUB$RCD3: allocated (CONTROL.FLOWERS.COM::DISK$CD:)

$ MOUNT/OVER=ID DISK$CONTROL_CD:
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-MOUNTED, VMS055LST1  mounted on _GRUB$RCD3:

$ DISMOUNT DISK$CONTROL_CD:

$ DEALLOCATE DISK$CONTROL_CD
$
```

This example allocates a CD-ROM for access between two OpenVMS systems. The drive is allocated, mounted, dismounted, and deallocated.

The next example allocates a CD-ROM drive on a remote machine running UNIX.

```
$ RMTALLOC /CD/NOWRITE SYS1:: DISK$SYS1_CD/USER=ROOT
%RMT-I-ALLOC, _GRUB$RCD3:  allocated (SYS1.FLOWERS.COM::/dev/rsr0)

$ MOUNT /OVER=ID DISK$MEL_CD:
%MOUNT-I-WRITELOCK, volume is write locked
%MOUNT-I-MOUNTED,  VMS055LST2  mounted on _GRUB$RCD3:

$ DISMOUNT DISK$MEL_CD:

$ DEALLOCATE DISK$MEL_CD:
$
```

This example allocates a UNIX CD drive. The device name defaults to /dev/rsr0. You could specify another device name, using the same example with the SYS1::"/dev/rsr42" value in the RMTALLOC command. After the device is allocated in the previous example, it is mounted, dismounted, and finally deallocated.

The next example allocates a tape and then invokes BACKUP to write to it.

```
$ REPLY /ENABLE

$ RMTALLOC COMMENT="PLEASE MOUNT TAPE #A1234" -
_$ WHORFIN::MKA500: TAPE
%%%%%%%%%% OPCOM  25-MAR-2017 11:24:35.46  %%%%%%%%%%%
(FROM NODE WHORFIN AT 25-MAR-2017 11:24:35.44)
REQUEST 87, FROM USER HOLMES ON WHORFIN
Please mount device _WHORFIN$mka500:
RMT tape  service request from WHORFIN.FLOWERS.COM
Please mount tape #A1234
 %%%%%% OPCOM  25-MAR-2017 11:25:29.12  %%%%%%%%%%%
(FROM NODE   HOLMES
25-MAR-2017 11:25:29.12)
REQUEST 87 WAS SATISFIED.
%RMT-I-ALLOC, _HOLMES$RMT2: ALLOCATED (WHORFIN.FLOWERS.COM::MKA500:)

$ INIT TAPE: FOO
$ BACKUP/LOG/INGORE=LABEL/VERIFY USERS:[ATMA.TEST]*.EXE;0 -
TAPE:EXES.BCK/SAVE
%MOUNT-I-MOUNTED, FOO MOUNTED ON _HOLMES$RMT2:
. .
$ BACKUP/LOG/IGNORE=LABEL/VERIFY USERS:[ATMA.TEST]*.H;0 -
TAPE:H.BCK/SAVE
. .
$ BACKUP/LOG/INGORE=LABEL/VERIFY USERS:[ATMA.TEST]*.C;0 -
TAPE:C.BCK/SAVE
. .
$ DISMOUNT/NOUNLOAD TAPE:

$ MOUNT/OVER=ID TAPE:
%MOUNT-I-MOUNTED, FOO MOUNTED ON _HOLMES$RMT2:

$ DIR TAPE:
DIRECTORY  _HOLMES$RMT2:[]
EXES.BCK;1     H.BCK;1       C.BCK;1
TOTAL OF  3 FILES.

$ DISMOUNT TAPE:

$ DEALL TAPE:
$
```

This example allocates access to a tape, then writes to it.

When issuing a RMTALLOC to a remote VSI TCP/IP system, the remote tape drive must be online with the tape physically loaded. Otherwise, RMTALLOC fails with the error, "%SYSTEM-F-MEDOFL, medium is offline."

You can override this default with the **/SEMANTICS=MOUNT** qualifier. RMTALLOC does not complete until a tape has physically been loaded and the tape drive is online. Use the **/SEMANTICS=COMMENT** keyword to specify a mount message to send to the operator via OPCOM.

IP RWALL

IP RWALL — Uses Remote Procedure Calls (RPCs) to send a network broadcast message to all users on the specified host. If you specify the host as an asterisk (*), the message is broadcast to all hosts on Ethernets to which the local host is attached.

Format

IP RWALL [*qualifier1*] [*qualifier2 . . .*] [*"message_text"*]

Restriction

RWALL messages are only received on hosts that support **RWALL** service.

Parameter

[*"message_text"*]

Contains the message to broadcast.

Qualifiers

/HEADER[=*"header_text"*]

/NOHEADER

Adds header text to the specified message. If you use the **/NOHEADER** qualifier, RWALL does not preface any header text to the specified message. By default, the header is prefaced with "Broadcast message from username@hostname:", although you may specify any header text as the value of this qualifier.

/HOST=[*hostname*]

Specifies the host on which the message is displayed. The default is **/HOST=LOCALHOST**, which prints the message on the host from which the RWALL command was invoked. If you specify the qualifier as **/HOST=***, the network broadcast displays on all directly reachable hosts on all connected networks that support broadcasting. **/HOST=*** is most appropriate for network-wide system shutdown messages.

Example

This example shows how to broadcast a shutdown message to users on the local host.

```
$ IP RWALL "Node ROMEO is shutting down"
RWALL MESSAGE:
Broadcast message from HOLMES@ROMEO: Node ROMEO is shutting down
```

IP SET /ARP

IP SET /ARP — Modifies Address Resolution Protocol (ARP) tables. These tables are normally modified dynamically by the ARP protocol. Use with **IP SHOW /ARP** to view the contents of the ARP table.

Qualifiers

/ADD=(PROTOCOL=*protocol*, HOST_ADDRESS=*host_addr*, ETHER_ADDRESS=*ether_addr*)

Adds a specified host-to-Ethernet address translation to the ARP tables. The PROTOCOL specification identifies which protocol (IP, for example) is being described. The HOST_ADDRESS specification gives the host address in IP form. The ETHER_ADDRESS specification gives the hardware Ethernet address in the form "aa:bb:cc:dd:ee:ff", where "aa" through "ff" are specified in hexadecimal. If not specified, the default is PROTOCOL=IP.

/COMMUNITY_NAME=*string*

Overrides the default community string (private) for remote SNMP **SET** requests. The **/SNMP_HOST** qualifier must be present if the **/COMMUNITY_NAME** qualifier is specified.

/DELETE=*host*

Deletes the specified host-to-Ethernet address translation from the ARP tables.

/FLUSH

Flushes the current ARP table. By default only temporary entries are flushed. If the qualifier /**PERMANENT** is specified, all entries are flushed.

/PERMANENT

/TEMPORARY (default)

Indicates that the translation to be added is kept (or deleted) permanently (used with the /**ADD** or /**FLUSH** qualifiers). The default (/**TEMPORARY**) indicates that this entry is considered for normal ARP table purging of old entries.

/PROXY

Used with the /**ADD** qualifier, indicates that the translation to the local host's Ethernet address is published on behalf of another host.

/PUBLISH

Indicates that the translation to be added is published on behalf of another host (that is, this host should answer with the specified translation on behalf of the other host). This qualifier is used with the /**ADD** qualifier.

/SNMP_HOST=*hostname*

Specifies the host affected by the **IP SET /ARP** command. The SNMP agent on the remote host must support read-write access to elements of the MIB-II variable `ipNetToMedia`.

Examples

This example displays the contents of the ARP table. Note: if the host name and IP address are longer than the "Host Network Address" field, they are truncated to fit.

```
$ IP SHOW /ARP /SYMBOLIC=NAMESERVER
VSI TCP/IP for OpenVMS ARP table:
Host Network Address                  Ethernet Address    Arp Flags
----------------------                ----------------    ---------
FSGATE.CC.FSTONE.COM (IP 128.0.33.123  AA:00:04:00:79:4C   Temporary
EXPLORER.ME.FSTONE.COM (IP128.0.41.1   08:00:11:00:90:B0   Temporary
GOOFY.CC.FSTONE.COM (IP 128.0.83.122)  08:00:20:01:27:6D   Temporary
BEGWS2.BEG.FSTONE.COM (IP 128.0.30.23  AA:00:04:00:65:4C   Temporary
ARPAGATEWAY.FSTONE.COM (IP 128.0.11.2  AA:00:04:00:0F:4C   Temporary
PORTAL1.CC.FSTONE.COM (IP 128.0.19.10  08:00:4C:00:23:CE   Temporary
WILMA.CC.FSTONE.COM (IP 128.0.7.125)   AA:00:04:00:64:4C   Temporary
FS4.CC.FSTONE.COM (IP 128.0.19.251)    AA:00:04:00:12:4C   Temporary
```

This example is often used to solve a problem that occurs in environments with a mixture of UNIX 4.2 BSD and 4.3 BSD systems. 4.2 BSD systems use zero-filled (nn.mm.0.0) IP broadcast addresses, while 4.3 BSD systems use ones-filled (nn.mm.255.255) broadcast addresses. To prevent 4.2 BSD systems from creating Ethernet "broadcast storms" when they issue ARP requests for the 4.3 BSD broadcast address, the above command publishes an ARP translation for the ones-filled broadcast address.

```
$ IP SET /ARP /ADD=(HOST=128.0.255.255,-
PROTOCOL=IP, ETHER=0:0:D:E:A:D) /PUBLISH
```

```
$
```

This example flushes all temporary ARP table entries.

```
$ IP SET /ARP /FLUSH
```

IP SET /DECNET

IP SET /DECNET — Configures the DECnet TCPA*x*: devices for running DECnet-over-UDP circuits.

Qualifiers

`/BUFFERS=`*`buffercount`*

Specifies the number of buffers the driver preallocates for this device (by default, 6).

`/CLOSE`

Shuts down and deletes a socket created with the **socket()** routine. After issuing a **CLOSE** command, the socket cannot be used again until the **IP SET /DECNET** command is reissued.

`/CONNECT`

Issues a **connect()** call to bind the remote address of the socket to the address specified in /**REMOTE_ADDRESS**.

`/DEVICE=`*`device`*

Specifies the DECnet device name (by default, TCPA0:).

`/FILTER_OUT_OF_ORDER=AUTOMATIC (default)`

`/FILTER_OUT_OF_ORDER=OFF`

`/FILTER_OUT_OF_ORDER=ON`

Controls the handling of out-of-order DECnet packets arriving via IP. Prior to OpenVMS V4.7, DECnet could not handle packets arriving out-of-order and would drop the line if it received them. If you have any OpenVMS V4.6 or earlier systems in your DECnet network with which you are communicating, you must use the **/FILTER_OUT_OF_ORDER=ON** qualifier. The default action, /**FILTER_OUT_OF_ORDER=AUTOMATIC**, selects the correct filtering based on the OpenVMS version of the current system only.

`/LOGDATA`

Specifies that **send()** and **recv()** log a sample of the data passed through them to OPCOM. Use this qualifier only for debugging network problems.

`/LOGERRORS`

Specifies that **send()** and **recv()** errors are logged to OPCOM. The default is to log all errors except these.

`/PORT=`*`UDP-port-number`*

Specifies the UDP port number to use for communication (by default, 700).

`/REMOTE_ADDRESS=`*`ip-address`*

Specifies the peer's IP address.

`/TCP=`*`mode`*

`/TCP=CONNECT`

`/TCP=LISTEN`

Specifies that DECnet is encapsulated in TCP instead of UDP. This mode is not supported by the normal configuration utility, but is of use over high-loss lines. LISTEN specifies that this end of the connection listens on the specified port; CONNECT specifies that this end attempts to connect to the listener on the specified port.

IP SET /IPS

IP SET /IPS — Controls the FILTER_SERVER process of the VSI TCP/IP Intrusion Detection and Prevention subsystem.

Qualifiers

`/DEBUG=`*`level`*

Specifies the level of debug for the filter server. Zero indicates no debug should be written to the log file, while increasing numbers indicate increasing amounts of debug will be written. This parameter should normally never be set above 4 without explicit instruction by VSI.

`/CLEAR_FILTERS`

Causes the `FILTER_SERVER` process to remove all filters set by IPS on all interfaces configured for IPS. This may be used with **SET /IPS /START** and **SET /IPS/RESTART**, or may be used by itself with **SET /IPS/CLEAR_FILTERS**. When used by itself this causes a running IPS subsystem to remove the IPS filters and reset the event count information for the source address associated with each filter being removed.

`/RELOAD`

Causes the filter server to re-read and parse the configuration files. Note that this will not wipe out existing event and rule information; it will simply update it so no potential filter information will be lost.

`/RESTART`

Stop and restart the filter server. All existing event and rule information will be lost and reloaded from the configuration files.

`/START`

Start the filter server if it’s not already running.

`/STOP`

Stop the filter server. All existing event and rule information will be lost.

IP SET /INTERFACE

IP SET /INTERFACE — Sets parameters for the specified network device. This command is invoked automatically by the network startup command file generated by the NET-CONFIG utility.

Format

`IP SET/INTERFACE` *`interface`*

Parameter

`interface`

Specifies the name of the interface to change; for example, "se0".

Qualifiers

`/ADDRESS=`*`network_address`*

Specifies a network address to assign to the network interface. The address format is dependent on the protocol specified with the **/PROTOCOL** specifier:

```
IP-address is of the form AA.BB.CC.DD IPX-address is a hexadecimal value
IPv6-address is of the form XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
```

`/ARP (default)`

`/NOARP`

/NOARP disables the Address Resolution Protocol on the specified interface (supported only on Ethernet interfaces).

`/COMMON_LINK=`*`line-ids`*

The **/COMMON_LINK** qualifier works for systems that have multiple interfaces on a common Ethernet, FDDI, or Token Ring cable. The system manager configures this support using the following qualifier:

```
$ IP SET /INTERFACE xxx/COMMON_LINK=(yyy[,zzz...])
```

xxx is the hardware device that the pseudo device that has the actual IP address of the machine is tied to (see the *VSI TCP/IP Administrator's Guide: Volume I* for an example on how to set up a pseudo device). yyy and zzz are device names like se0, se1, and se2. With this qualifier, VSI TCP/IP links the interfaces together. A performance benefit of this linking occurs if data is to be transmitted on an interface that happens to be busy, VSI TCP/IP assigns the data to the least busy linked interface for transmission.

This linking also provides a level of redundancy. If a linked interface is shut down using **IP SET / INTERFACE/DOWN** or if a fatal error is detected with the interface and an automatic restart can not be attempted, then any routing table entries or pseudo devices associated with the shut down interface will be failed over to one of the common link interfaces.

Restrictions:

- The joined interfaces must be connected to the same cable.
- The joined interfaces must have the same MTU.

Example Configuration:

The actual IP address for SYSA.EXAMPLE.COM is 192.168.0.1; this address is used for a pseudo device (pd0), which uses se0.

```
$ IP configure/network
VSI TCP/IP for OpenVMS Network Configuration Utility V10.5(104)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>show
Interface                        Adapter CSR Address  Flags/Vector
---------                        ------- -----------  ------------
se0      (Shared OpenVMS Ethernet/FDDI) -NONE-  -NONE-       -NONE-
         [TCP/IP$: 192.168.1.1, IP-SubNet: 255.255.255.0]
         [VMS Device: EWA0, Link Level: Ethernet]
se1      (Shared OpenVMS Ethernet/FDDI) -NONE-  -NONE-       -NONE-
         [TCP/IP$: 192.168.1.2, IP-SubNet: 255.255.255.0]
         [VMS Device: EWB0, Link Level: Ethernet]
pd0      (Secondary Ethernet Address)   -NONE-  -NONE-       -NONE-
         [TCP/IP$: 198.168.0.1, IP-SubNet: 255.255.255.0]
         [Hardware-Device: se0]
Official Host Name:             sysa.example.com
Domain Nameserver:              127.0.0.1
Timezone:                       EST
Timezone Rules:                 US/EASTERN
Load UCX $QIO driver:           TRUE
Load PWIP (Pathworks) driver:   TRUE
SNMP Agent X subagents are enabled
NET-CONFIG>
```

If DECnet is being used, then IP$:SE1_CONFIGURE.COM will need to be created (see IP$ROOT:[IP.EXAMPLES]SE0_CONFIGURE.COM) to configure SE1 without the DECnet Ethernet address.

The command **$ IP SET /INTERFACE SE0/COMMON_LINK=(SE1)** can be added to the **SE1_CONFIGURE.COM**, or put in IP$:LOCAL_INITIALIZATION.COM so that it will be executed each time VSI TCP/IP is started.

/COMMUNITY_NAME=*string*

Overrides the default community string (private) for remote SNMP SET requests. The /**SNMP_HOST** qualifier must be present if the **/COMMUNITY_NAME** qualifier is specified.

/CREATE

Requests that a dynamic interface (e.g. gif1) be created

/DELETE

Requests that a dynamic interface (e.g. gif) be deleted.

/D1

`/NOD1 (default)`

Enables or disables the device-dependent IFF_D1 flag.

`/D2`

`/NOD2 (default)`

Enables or disables the device-dependent IFF_D2 flag.

`/D3`

`/NOD3 (default)`

Enables or disables the device-dependent IFF_D3 flag.

`/DEBUG`

`/NODEBUG (default)`

Enables interface-specific debugging. Some interfaces have debugging code and send debugging information to the users with OPCOM OPERATOR messages enabled.

`/DECNET_ETHERNET_ADDRESS (default)`

`/NODECNET_ETHERNET_ADDRESS`

Initializes a DECnet shared Ethernet interface to determine what Ethernet address to use. If other protocols are currently using the device, the Ethernet address cannot be changed and this qualifier is ignored.

The default behavior, **/DECNET_ETHERNET_ADDRESS**, is used by **IP SET /INTERFACE** to look at the *`SCSSYSTEMID SYSGEN`* parameter and set the Ethernet address to match. If SCSSYSTEMID is not set, the address on the Ethernet card's PROM is used.

If **/NODECNET_ETHERNET_ADDRESS** is specified, VSI TCP/IP uses the PROM address.

`/DOWN`

Marks the network interface as not UP and packets are no longer accepted or transmitted. See **/UP (default)** for more information.

`/DYNAMIC`

`/NODYNAMIC (default)`

Reverts the terminal line to a normal OpenVMS terminal line if a modem hangup occurs. Use **/DYNAMIC** to create dynamic-dialup SLIP links with the **/LINK_LEVEL=SLIP** qualifier. When creating a dynamic SLIP link, CMKRNL, LOG_IO, and SYSPRV privileges are required.

`/FFI_BUFFERS=`*`number_of_buffers`*

Initializes a shared OpenVMS Ethernet or FDDI interface, and specifies the number of packet buffers to allocate to each protocol port of the OpenVMS device driver (by default, 4).

`/FILTER=`*`filter_file`*

`/NOFILTER (default)`

Associates a file containing a packet filter list with a particular network interface. The contents of this file are parsed and the individual filters are loaded for the interface. If the file `IP$:FILTER-`*`interface`*`.DAT` exists when VSI TCP/IP is started, the VSI TCP/IP startup procedure will automatically load these filters for the specific interface.

`/FORMAT=[NORMAL | COMMA]`

Log events in the specified format. If NORMAL, then the formatting used by **IP SHOW / INTERFACE /FILTER** is used. If COMMA, then a comma-delimited line is output to the file. This can then be loaded into, for example, a spreadsheet for analysis. If the log destination is OPCOM, use of the **/FORMAT** qualifier is illegal.

`/HARDWARE_DEVICE=`*`primary_interface`*

Specifies the name of the real interface for a secondary IP address device, and connects the interface to the specified primary interface.

`/INTERVAL=`*`seconds`*

Reporting interval in seconds. The minimum reporting interval is seconds, so that a flood of filter events doesn't adversely impact the system. The minimum interval that can be specified is 5 seconds. If no interval has been specified when logging is enabled (see the **/LOG** qualifier in this section), an interval of 5 seconds will be used.

`/IP_BROADCAST=`*`ip_address`*

Specifies a non-standard IP broadcast address. The default IP broadcast address has all bits in the host part of an IP address set to 1 (the standard format under 4.3 BSD). Some sites may still use the 4.2 BSD standard of IP broadcasts with the host part of an IP address set to 0.

`/IP_SUBNET_MASK=`*`ip_address`*

Specifies the network portion of the interface IP address. ip_address is an IP address in which each bit corresponding to a bit in the network portion is set to 1. All interfaces on the same subnet must have the same subnet mask.

By default, VSI TCP/IP uses the subnet mask implied by the interface's IP address. Do not use the default subnet mask if your site has subnets. For example, the default subnet mask of an interface with the address 161.44.128.15 is 255.255.0.0. (255.255.255.0 would be a suitable subnet mask if that interface is on a subnet, and there are fewer than 256 subnets, and the total number of hosts is less than 256.)

`/IP6_SUBNET_MASK=`*`length`*

Specifies the length of the IPv6 subnet mask. The range of this is from 1 to 128 bits. The default length is 128.

/LINK_LEVEL= ethernet, ppp, proteon, slip

`/LINK_LEVEL=`*`ethernet, ppp, proteon, slip`*

Specifies the type of device being initialized. Use **/LINK_LEVEL** with the **/VMS_DEVICE** qualifier. This qualifier supersedes the former **/SLIP_DEVICE** and **/PROTEON_DEVICE** qualifiers.

- Specify PPP for Point-to-Point Protocol devices.
- Specify SLIP for Serial Line Internet Protocol (SLIP) devices.
- Specify ETHERNET for ETHERNET_II encapsulation.

`/LOCAL=`*`node_name`*

Specifies the name of the local node on this side of an IP interface; may be used with DECnet and PSI links.

`/LOG=[`*`filename`*` | OPCOM]`

Used to turn logging on or off for those filters that contain the LOG qualifier in their definition. The logging may be to OPCOM or the specified file. Turn logging off using /**NOLOG**.

`/MTU=`*`mtu`*

Specifies the Maximum Transmission Units-the size of IP packets over a given interface. Not all devices support the use of /**MTU**, and there may be additional, device-dependent restrictions dictating when it can be used.

`/MULTICAST=ALL`

Enables reception of all multicast packets.

`/PEER=`*`peer_name`*

Specifies the name of the node on the other side of an IP interface; used with DECnet and PSI links.

`/POINT_TO_POINT_DESTINATION=`*`ip_address`*

Specifies the IP address of the node on the other side of a point-to-point interface.

`/PPP_NOICMP`

Prevents ICMP packets from being passed to IP via the PPP interface.

`/PPP_OPTIONS=`*`options_list`*

Specifies values for the PPP options included in a comma-separated option_list. The following options may be enabled:

ACCM=*mark*	MRU=*size*
AUTHENTICATION=*method*	NOICMP
COMPRESS_PROTOCOL	TCP_COMPRESSION
COMPRESS_ADDRESS_AND_CONTROL	TERMINATION_RETRIES=*count*
CONFIGURATION_RETRIES=*count*	TIMEOUT=*seconds*
IDLE=*seconds*	

`/PROTOCOL=`*`protocol_name`*

Specifies the protocol to which the /**ADDRESS** qualifier refers (by default, IP). For IPv6 use I6.

`/PREFIX=`*`ipv6_prefix`*

Specifies the IPv6 prefix for an interface to use to generate a global IPv6 address. The default prefix length is 64, or a different value can be specified with the **IP6_SUBNET_MASK** qualifier.

`/RARP`

`/NORARP (default)`

Initializes the OpenVMS Ethernet device to receive RARP packets. The **/RARP** qualifier is used with the **/VMS_DEVICE** qualifier. The RARP packet type is disabled by default and must be enabled to use the RARP service on OpenVMS Ethernet devices.

`/SEND_QUEUE_LENGTH=`*`number`*

Specifies the maximum queue length for packets waiting to be sent from the interface. The minimum value is 10, default values are interface specific. If an interface has a heavy transmit load and is showing dropped packets, then specifying a larger number here may help.

`/SNMP_HOST`

Specifies the host affected by the **IP SET /INTERFACE** command. The SNMP agent on the remote host must support read-write access to the MIB-II variable `ifAdminStatus`.

/SNMP_HOST can only be used with the /UP or /DOWN qualifiers.

The device specified with the **/SNMP_HOST** qualifier may be either the full text string of the remote interface name or the numeric index of the interface to be set. You can display a list of remote interface names with the **IP SHOW /INTERFACE /SNMP_HOST** command.

`/TRAILERS`

`/NOTRAILERS (default)`

Enables IP trailer encapsulation for the specified interface (only supported on Ethernet and FDDI interfaces). If trailers are enabled, the use of IP trailer encapsulation is negotiated between hosts as a byproduct of IP-to-Ethernet address resolution using Address Resolution Protocol (ARP). On an HP Ethernet controller, **/TRAILERS** must be used with /VMS to initialize the trailer protocol ports.

`/TUNNEL=(DESTINATION_ADDRESS=`*`ip_address`*`, GATEWAY_ADDRESS=`*`ip_address`*`)`

Set up a tunnel with a gif interface. Specifies the local (gateway) and remote (destination) public addresses when setting tunnel addresses. Tunnels also need a local address set with **/ADDRESS** and a remote address set with **/POINT_TO_POINT_DESTINATION**. For more detail see the *VSI TCP/IP Administrator's Guide: Volume II.*

`/UP (default)`

`/DOWN`

/UP marks the network interface as "up" and ready to accept or transmit packets. **/DOWN** marks the network interface "down" and packets are no longer accepted or transmitted.

`/VMS_DEVICE=[`*`vms_device`*`]`

Initializes an interface that has an associated OpenVMS device, telling the VSI TCP/IP kernel which OpenVMS device to associate with the IP device. If **/VMS_DEVICE** is used with **/DOWN**, the specified OpenVMS device is disconnected from the IP device and made available to other OpenVMS applications.

Examples

This example disables the se0 interface.

```
$ IP SET/INTERFACE SE0 /DOWN
```

This example enables the se0 interface with the address 192.0.0.1.

```
$ IP SET/INTERFACE SE0 /UP/ADDRESS=192.0.0.1
```

This example enables a dynamic SLIP line.

```
$ IP SET/INTERFACE SL1 /DYNAMIC/LINK_LEVEL=SLIP/VMS_DEVICE
```

Enter the following command at VSI TCP/IP startup:

```
$ IP SET/INTERFACE PD0/COMMON_LINK=(SE0,SE1)
```

The PD0 has the real IP address, the SE*n* devices have something else (like 10.*n.n.n*).

```
$ IP SET/INTERFACE SE0 /LOG=OPCOM/INTERVAL=10
```

enables logging to OPCOM, with a reporting interval of 10 seconds.

```
$ IP SET/INTERFACE SE0 /LOG=FOO.DAT/FORMAT=COMMA
```

enables logging to the file FOO.DAT in comma-delimited format, and a reporting interval of 5 seconds (the default).

```
$ IP SET /INTERFACE SE0 /NOLOG
```

This disables all logging for the interface, closing all open log files.

IP SET /ROUTE

IP SET /ROUTE — Specifies static IP routing, including the default route. This command is invoked automatically by the network startup command file generated by the Network Configuration Utility (NET-CONFIG). Before making changes with **SET /ROUTE**, use **IP SHOW /ROUTE** to view the routing information.

Qualifiers

/ADD=(DESTINATION=*ip-address*,GATEWAY=*ip-address* [,NETMASK=*network-mask*] [,INTERFACE][,MASK_LENGTH=*integer*])

Adds a static IP route to the VSI TCP/IP kernel routing tables.

- The DESTINATION specification gives the NETWORK or HOST for which the routing information is valid.
- The GATEWAY specification gives the next hop for the packet to take on its way to the DESTINATION.

- The optional INTERFACE keyword forces the routing to be for a locally connected interface, and is normally not used.
- The optional NETMASK specification dictates which bits of the DESTINATION IP-address comprise the network portion of an IP-address. If not specified, the DESTINATION address is given a class-based network mask.
- The optional MASK_LENGTH specifies the length in bits of the mask to apply to the DESTINATION address. Either NETMASK or MASK_LENGTH can be specified, not both.

`/COMMUNITY_NAME=`*`string`*

Overrides the default community string (private) for remote SNMP SET requests. The **/SNMP_HOST** qualifier must be present if the **/COMMUNITY_NAME** qualifier is specified.

`/DELETE=(DESTINATION=`*`ip-address`*`, GATEWAY=`*`ip-address`*
`[,NETMASK=`*`network-mask`*`] [,INTERFACE])`

Deletes an IP route from the VSI TCP/IP kernel routing tables.

- The DESTINATION specification gives the NETWORK or HOST for which the routing information is valid.
- The GATEWAY specification gives the next hop for the packet to take on its way to the DESTINATION.
- The optional INTERFACE keyword forces the routing to be for a locally connected interface, and is normally not used.
- The optional NETMASK specification dictates which bits of the DESTINATION IP-address comprise the network portion of an IP-address. If not specified, the DESTINATION address is given a class-based network mask.

`/FLUSH`

Deletes all IP routes in the VSI TCP/IP kernel.

`/FORCE_HOST`

Interprets the DESTINATION as a HOST address when used with the **/ADD** or the **/DELETE** qualifiers.

`/FORCE_NETWORK`

Interprets the DESTINATION as a NETWORK address when used with the **/ADD** or the **/DELETE** qualifiers.

`/NETWORK_IMAGE=`*`file-spec`*

Specifies the network image associated with the running VSI TCP/IP kernel. This is used to read IP routing information in the VSI TCP/IP kernel. If not specified, the image currently loaded is used.

`/PROTOCOL=`*`protocol_name`*

Specifies the protocol that the route applies to. The default is IP, use I6 for IPv6.

`/SNMP_HOST=`*`hostname`*

Specifies an IP host. The SNMP agent on the remote host must support read-write access to elements of the MIB-II variable `ipRouteTable`.

Examples

This example displays the current state of the VSI TCP/IP routing tables. **/NOSYMBOLIC** forces **IP SHOW /ROUTE** to display the information numerically.

```
$ IP SHOW /ROUTE /NOSYMBOLIC
VSI TCP/IP for OpenVMS IP Routing tables:
Destination    Gateway      Flags        Refcnt   Use     Interface
------------   -------      ------       ------   ---     ---------
127.0.0.1      127.0.0.1    Up,Host      2        2529    lo0
192.0.0.1      192.0.0.2    Up,Host      3        10521   sl0
0.0.0          192.0.0.1    Up,Gateway   3        6105    sl0
192.0.0.64     192.0.0.65   Up           2        2372    se0
```

This example deletes the default route to `FLOWERS.COM`.

```
$ IP SET/ROUTE/DELETE=(DEST=DEFAULT,GATE=192.0.0.1)
Delete Route DEFAULT, Gateway FLOWERS.COM
$
```

IP SET /TIMEZONE

IP SET /TIMEZONE — Specifies the local timezone name that was either previously compiled into VSI TCP/IP or is a name from a selected timezone in the timezone database files.

Format

`IP SET /TIMEZONE` *`localzone`*

Parameter

`localzone`

The name of the local timezone; for example, "PST."

Qualifiers

`/LOG`

`/NOLOG (default)`

Displays a list of the timezones that are loaded, and a list of the compiled-in zones that were selected but not loaded because they were compiled in.

`/SELECT=(`*`rule1`*` [,`*`rule2`*` [...]])`

Specifies a list of countries or timezones to load. Specifying a country loads all timezones in that country.

`/FILES=(`*`file1`*` [,`*`file2`*` [...]])`

Specifies a list of files from which to load the timezone data. The default is `IP$:TIMEZONES.DAT`. Locally-written rules are normally added to `IP$:TIMEZONES.LOCAL`.

Examples

This example sets the local timezone to PST.

```
$ IP SET /TIMEZONE PST
```

This example sets the local timezone to MST and loads Arizona timezone rules.

```
$ IP SET /TIMEZONE MST/SELECT="US/ARIZONA"
```

IP SETKEY

IP SETKEY — Manually manipulates the IPsec SA/SP database. In order to use SETKEY, a foreign command needs to be defined. **$ SETKEY :== $IP$:SETKEY.EXE** Note that only UNIX-style options can be used. For more details, please refer to the *VSI TCP/IP Administrator's Guide: Volume II*.

Synopsis

```
setkey [-v] -c
```

```
setkey [-v] -f filename
```

```
setkey [-aPv] -D
```

```
setkey [-Pv] -F
```

```
setkey [-h] -x
```

Description

SETKEY adds, updates, dumps, or flushes Security Association Database (SAD) entries, as well as Security Policy Database (SPD) entries in the kernel.

SETKEY takes a series of operations from the file named `IP$: IPSEC.CONF` (when invoked with -f *`filename`*).

Arguments

Note

Since **SETKEY** supports both uppercase and lowercase command options, these have to be enclosed within quotation marks (e.g, setkey "-F").

`-a`	Also displays the SAD (Security Association Database) entries. A SAD entry is when it has expired, but it may still be referenced by SPD (Security Policy Database) entries.
`-D`	Dumps the SAD entries. If used with -P, the SPD entries are dumped.

`-F`	Flushes the SAD entries. If used with -P, the SPD entries are flushed.
`-xx`	Makes each timestamp unformatted.
`-h`	Adds hexadecimal dump on -x mode.
`-l`	Loops forever with short output on -D.
`-P`	Dumps (when specified with -D) or flush (with -F) the SPD entries.
`-v`	Verbose. The program will dump messages exchanged on PF_KEY socket, including messages sent from other processes to the kernel.
`-x`	Loops forever and dumps all the messages transmitted to the PF_KEY socket.
`-f` *`filename`*	File that contains the operations to be performed. For more information about the operations, see the section called "Header Operations" section.

Header Operations

Header Operations have the following grammar. Note that lines starting with hashmarks ('#') are treated as comment lines.

`add` *`src dst`* protocol *`spi`* [extensions] *`algorithm`*;	Adds a SAD entry.
`get` *`src dst protocol spi ;`*	Shows a SAD entry.
`delete` *`src dst protocol spi ;`*	Removes a SAD entry.
`delete all` *`src dst protocol ;`*	Removes all SAD entries that match the specification.
`flush` [*`protocol`*];	Clears all SAD entries matched by the protocol.
`dump` [*`protocol`*];	Dumps all SAD entries matched by the protocol.
`spdadd` *`src_range dst_range upperspec policy;`*	Adds an SPD entry.
`spddelete` *`src_range dst_range upperspec`* `-P` *`direction ;`*	Deletes an SPD entry.
`spdflush;`	Clears all SPD entries.
`spddump;`	Dumps all SPD entries.

Meta-arguments

Meta-arguments used in the header operations are as follows:

`src dst`

Source/destination of the secure communication is specified as an IPv4 address. **setkey** does not consult hostname-to-address for arguments ***`src`*** and ***`dst`***. They must be in numeric form.

`protocol`

`protocol` is one of following:

esp	ESP based on rfc2405
ah	AH based on rfc2402

`spi`

Security Parameter Index (SPI) for the SAD and the SPD. It must be decimal number or hexadecimal number You cannot use the set of SPI values in the range 0 through 255. (with 0x attached).

`extensions`

Take some of the following:

`-m` *`mode`*	Specifies a security protocol mode for use. *`mode`* is one of following: transport, tunnel or any. The default value is any.
`-E` *`ealgo key`*	Specifies an encryption algorithm.
`-A` *`aalgo key`*	Specifies an authentication algorithm. If -A is used with *`protocol`* esp, it will be treated as ESP payload authentication algorithm.*`protocol`* esp accepts -E and -A. *`protocol`* accepts -E only. *`protocol`* ah accepts **-A** only

`key` must be double-quoted character string or series of hexadecimal digits. Possible values for *`ealgo`*, *`aalgo`* and *`calgo`* are specified in separate section.

`src_range dst_range`

These are selections of the secure communication specified as IPv4/v6 address or IPv4/v6 address range, and it may accompany TCP/UDP port specification. This takes the following form:

`address`

`address/prefixlen`

`address[port]`

`address/prefixlen[port]`

`prefixlen` and *`port`* must be decimal number. The square bracket around *`port`* is really necessary. They are not manpage metacharacters.

SETKEY does not consult hostname-to-address for arguments *`src`* and *`dst`*. They must be in numeric form.

`upperspec`

Upper-layer protocol to be used. “icmp” and “any” can be specified. “any “stands for ``any protocol''. You can also use the protocol number.

Note

`upperspec` does not work against forwarding case at this moment, as it requires extra reassembly at forwarding node (not implemented at this moment). There are many protocols in *`/etc/protocols`*, but protocols other than TCP, UDP, and ICMP may not be suitable to use with IPSec.

`policy`

policy is the one of following:

`-P`

`direction discard`

`-P`

`direction none`

`-P`

`direction ipsec protocol /mode /src-dst /level`

You must specify the policy's direction as *`direction`* by using either “out” or “in”.

`discard` means the packet matching indexes will be discarded.

`none` means that IPsec operations will not take place onto the packet.

`ipsec` means that IPSEC operation will take place onto the packet. “ah,” “esp” or “ipcomp” must be set as *`protocol`*.

`mode` is either transport or tunnel. If *`mode`* is tunnel, you must specify the end-point addresses of the SA as *`src`* and *`dst`* with `-' between these addresses, which is used to specify the SA. If *`mode`* is transport, both *`src`* and *`dst`* can be omitted. *`level`* is to be one of the following: “default”, “use”, “require” or “unique”. If the SA is not available in every level, the kernel will request getting the SA to the key exchange daemon. “default” means the kernel consults to the system wide default against protocol you specified, e.g. `esp_trans_deflev sysctl` variable, when the kernel processes the packet. “use” means that the kernel uses an SA if it's available, otherwise the kernel keeps normal operation. “require” means an SA is required whenever the kernel sends a packet matched with the policy.

“unique” is the same as “require”, except that “unique”allows the policy to bind with the unique outbound SA. If you use the SA by manual keying, you can put the decimal number as the policy identifier after “unique”, provided it is separated by a colon `' similar to this example: unique:*`number`*. *`number`* must be between 1 and 32767. It corresponds to *`extensions`* **-u**.

Algorithms

The following list shows the supported algorithms. Following is a list of authentication algorithms that can be used as *`aalgo`* in -A of the *`protocol`* parameter:

```
algorithm       keylen (bits)   comment
hmac-md5        128             ah: rfc2403
     128            ah-old: rfc2085
hmac-sha1       160             ah: rfc2404
     160            ah-old: 128bit ICV (no document)
keyed-md5       128             ah: 96bit ICV (no document)
     128            ah-old: rfc1828
keyed-sha1      160             ah: 96bit ICV (no document)
     160            ah-old: 128bit ICV (no document)
null            0 to 2048       for debugging
hmac-sha2-256   256             ah: 96bit ICV (no document)
```

```
        256            ah-old: 128bit ICV (no document)
   hmac-sha2-384   384            ah: 96bit ICV (no document)
        384            ah-old: 128bit ICV (no document)
   hmac-sha2-512   512            ah: 96bit ICV (no document)
        512            ah-old: 128bit ICV (no document)
```

Following is a list of encryption algorithms that can be used as *`ealgo`* in -E *`ealgo`* of *`protocol`* parameter:

```
algorithm      keylen (bits)   comment
des-cbc        64              esp-old: rfc1829, esp: rfc2405
3des-cbc       192             rfc2451
blowfish-cbc   40 to 448       rfc2451
cast128-cbc    40 to 128       rfc2451
```

SETKEY File Example

```
add    10.0.11.41 10.0.11.33 esp 0x110010
               -E des-cbc “ESP with”
               -A hmac-md5 “authentication!!” ;
      flush ;
     dump esp ;
     spdadd  10.0.11.41/32[21] 10.0.11.33/32[any] any
               -P out ipsec esp/transport/192.168.0.1-192.168.1.2/require ;
```

IP SHOW

IP SHOW — Displays VSI TCP/IP network information.

Format

```
IP SHOW
```

Qualifiers

`/ALL`

Displays information provided by all other **IP SHOW** qualifiers.

`/ARP`

Displays the Address Resolution Protocol (ARP) tables.

`/BUFFERS`

Displays VSI TCP/IP kernel memory usage statistics.

`/COMMUNITY_NAME=`*`community`*

Overrides the default community string (public) for remote SNMP requests. The **/SNMP_HOST** qualifier must be present if the **/COMMUNITY_NAME** qualifier is specified. A value must be passed to this qualifier.

`/CONFIGURATION`

Displays network interface configuration information.

/CONNECTIONS[=

[*all*]

[*pid*]

[*process_names*]

[*nokernel*]

Displays network connections.

- If you specify **IP SHOW** with no qualifiers, **/CONNECTIONS** is the default.
- If you specify the ALL keyword, sockets associated with active listeners also display.
- If you specify the PID keyword, the process ID (PID) displays.
- If you specify the PROCESS_NAMES keyword, the name of the process that owns each socket displays. Sockets not associated with a process (for example, an inbound TELNET session) display with a process name of kernel.
- If you specify PID or PROCESS_NAMES and the NOKERNEL keyword, connections not associated with processes do not display.

Note

Line information is truncated if the display width is too small. As a consequence, IP addresses may appear incomplete. To display more complete information, increase the display width with the **SET TERM /WIDTH=[*value*]** or **IP SHOW /CONNECTIONS/WIDTH=[*value*]** at the command prompt.

/CONTINUOUS

Updates the display continuously with information about the network by using the OpenVMS Screen Management Graphics (SMG) library routines. If used with more than one other qualifier, **IP SHOW** cycles between the different displays.

/IPS

When used with the **/CONFIG=*filename*** qualifier, writes the current stats of the filter server to the specified filename.

/FULL

Displays more information about a queue. Use **/FULL** only with **/QUEUE**. (See **/QUEUE=*queue_name*** for more information.)

/HOST

Displays addresses and names for the host name specified.

/INTERFACE

Displays information about a specific interface. Use the **IP SHOW /STATISTICS** command to display the available interfaces, then use **SHOW /INTERFACE** to display additional information on each interface.

`INTERFACE/EXTRACT_FILTERS=`*`file`*

Reads all non-expired filters from the specified interface and writes them to the specified filename in the same text format that would be used as input to the **IP SET/INTERFACE/FILTER** command.

`/IP`

Shows network connections. (/**IP** is the same as /**CONNECTION**.)

`/MIB_VAR=`*`mibIIvalue`*

Displays the value of SNMP MIB variables; used with the **/SNMP_HOST** qualifier. This value can be any MIB II variable described in RFC-1213.

`/LICENSE[=`

`[`*`all`*`]`

`[`*`IP`*`]`

`[`*`nfs_server`*`]`

`[`*`nfs_client`*`]`

Displays the status of VSI TCP/IP software product licenses. Without a keyword, this qualifier displays license information including the authorization for VSI TCP/IP products. The `ALL` keyword is the default. All other values display license status for the specified product.

`/NFSMOUNT[=`*`all`*`], [`*`directory`*`], [`*`exports`*`]`

Indicates which hosts are mounted on your system, and what mount points are exported by the server.

- `ALL` displays all remote mounts.
- `DIRECTORY` displays directories that have been remotely mounted by clients.
- `EXPORTS` displays a list of exported file systems.

Use /**NFSMOUNT** with /**REMOTE** to display information about a remote host.

`/OUTPUT=`*`file_spec`*

Specifies a filename to which the command output is written. The default is `SYS$OUTPUT`.

`/PROTOCOLS=`*`all, internet, ip, ipx, ns, spx, tcp`*

Specifies the protocols about which information is displayed. The default, /**PROTOCOLS=ALL**, displays information about all active protocols. Use /**PROTOCOLS** with other qualifiers. The quantity of information displayed varies by queue hardware; for example, the UNIX operating system shows more than just queues handled by other independent vendor's queue controllers.

`/QUEUE=`*`queue_name`*

`/full`

`/nofull (default)`

Displays the contents of the specified local OpenVMS and corresponding remote LPD protocol queues. Use the TCP LPD service to access the contents of the remote queue for display. If **/FULL** is specified, the queue is displayed in long form. If the remote system is also running VSI TCP/IP, the long form is identical to the short form. VSI TCP/IP queues configured with the STREAM protocol cannot be displayed with this command.

`/REMOTE_HOST=`*`host`*

Displays network status and configuration information about a remote host by using the NETSTAT service. The host specification can be either a host name or address. The remote host must support the NETSTAT service for this command to work.

If the remote host is also a VSI TCP/IP system, this command is the same as the **IP SHOW /ALL** on the remote host.

`/ROUTE`

[*`/destinations=(dest1[,dest2, . . . ]`*]

[*`/gateways=(gateway1[,gateway2, . . . ]`*]

[*`/interfaces=(interface1[,interface2, . . . ])`*]

Displays routing information for the IP, IPX, NS, and SPX protocols.

- **/DESTINATIONS** displays only routes to these destination addresses; this qualifier is only valid for IP routes.
- **/GATEWAYS** displays only routes through these gateways; this qualifier is only valid for IP routes.
- **/INTERFACES** displays only routes through these interfaces.

You can use all other **IP SHOW** qualifiers with **IP SHOW /ROUTE**.

The **/ROUTE** qualifier must precede all other qualifiers.

`/RPC_PORTMAP`

Displays the currently registered RPC protocols by contacting the RPC portmapper.

`/STATISTICS[=`*`interface`*`], [`*`protocol`*`], [`*`all`*`]`

Displays network interface statistics, protocol statistics, or both. If **/STATISTICS** is specified with no value, interface statistics are displayed.

`/SNMP_HOST=`*`hostname`*

Used with the following **IP SHOW** qualifiers to obtain information from a remote SNMP agent. You can override the default community name (public) using the **/COMMUNITY_NAME** qualifier.

`/COMMUNITY_NAME`

/CONNECTIONS[=(all)]

/ARP

/MIB_VAR

/ROUTE

Note

/ROUTE must precede **/SNMP_HOST** on the command line.

/STATISTICS *host_table (default)*

/SYMBOLIC_ADDRESSES =*nameserver*

/NOSYMBOLIC_ADDRESSES *nameserver_first*

Determines how certain fields in the output are formatted before being displayed to the user. These qualifiers are used with the other **IP SHOW** qualifiers.

- **/SYMBOLIC_ADDRESSES=HOST_TABLE** specifies that the static host tables are used to translate IP addresses to host names, network numbers to network names, and port numbers to service names.
- **/SYMBOLIC_ADDRESSES=NAMESERVER** specifies that the Domain Name System (DNS) is queried to translate IP addresses into host names if the normal host table lookup fails. This operation can generate many queries to DNS domain servers (and can, therefore, be quite slow).
- **/SYMBOLIC_ADDRESSES=NAMESERVER_FIRST** specifies that the DNS is queried first to translate IP addresses into host names, falling back to the host tables if the query should fail.
- **/NOSYMBOLIC_ADDRESSES** specifies that "raw" protocol addresses and port number are displayed in the output, rather than determining the host, network, and service names that correspond to the addresses and numbers.

/TCP

Shows network connections. (**/TCP** is the same as **/CONNECTION**.)

/VERSION

Displays the VSI TCP/IP version and the version of the OpenVMS Operating System.

/WIDTH=*width*

Specifies the width of displayed output when used with the **/ARP**, **/CONNECTIONS**, **/ROUTE**, and **/STATISTICS** qualifiers. The width must be greater than 80.

Examples

This example shows how to use the **/OUTPUT** qualifier to direct the output of a **IP SHOW** command to the file IP.ALL.

```
$ IP SHOW /ALL /OUTPUT=IP.ALL
$ IP SHOW
```

```
VSI TCP/IP Active Connections:
Proto  Rcv-Q  Snd-Q Local Address (Port) Foreign Address  State
-----  -----  ----- -------------------- ---------------  -----
TCP        0      0 LOCALHOST(790)       LOCALHOST(RPC)   TIME_WAIT
TCP        0      0 LOCALHOST(1033)      LOCALHOST(SMTP)  TIME_WAIT
TCP        0      0 FLOWERS(NETSTAT)     WARBUCKS(3335)   FIN_WAIT_2
TCP        0      0 FLOWERS(FTP)         WARBUCKS(3334)   ESTABLISHED
TCP        0      0 FLOWERS(1031)        WARBUCKS(TELNET) ESTABLISHED
UDP        0      0 FLOWERS(NAMESERV)    *(*)
UDP        0      0 LOCALHOST(NAMESERV)  *(*)
UDP        0      0 FLOWERS(DECNET)      IU(DECNET)

$ IP SHOW /CONFIGURATION
** Configuration for file "IP$:NETWORK_DEVICES.CONFIGURATION" **
Device                       Adapter   CSR Address   Flags/Vector
------                       -------   -----------   ------------
se0  (Shared OpenVMS Ethernet)   -NONE-    -NONE-        -NONE-
s10  (Serial Line IP)        -NONE-    -NONE-        -NONE-
dn0  (IP over DECNet link)   -NONE-    -NONE-        -NONE-

$ IP SHOW /STATISTICS=INTERFACE
VSI TCP/IP Network Interface statistics:
Name   Mtu    Network       Address       Ipkts   Ierrs   Ipkts   Oerrs
 Collis
----   ----   ----------    ----------    -----   -----   -----   -----
 ------
se0    1500   FLOWERS-NET   FLOWERS.COM   150     0       116     0       0
s10    1006   FLOWERS-NET   FLOWERS.COM   597     0       697     0       0
pd0    1500   FLOWERS-NET   192.0.0.1     0       0       0       0       0
dno*   1500   FLOWERS-NET   FLOWERS.COM   0       0       0       0       0
lo0    1536   LOOPBACK-NET  LOCALHOST     53      0       53      0       0
$
```

This example displays the status of VSI TCP/IP licenses.

```
$ IP SHOW /LICENSE
VSI TCP/IP for OpenVMS, HP rx2600  (1.30GHz/3.0MB), OpenVMS I64 V8.2-1
Product      License   Authorization
-------      -------   -------------
IP           Yes       A-2336-15873
NFS-SERVER   Yes       A-2336-15879
NFS-CLIENT   Yes       A-2336-15882
$
```

In this example, user ROSE on host FLOWERS.COM has issued a print request to print the file PROGRAMMERS.PS on the REMOTE_PS local queue. The REMOTE_PS queue, however, is a VSI TCP/IP remote print queue that uses the LPD protocol to send the print request to the print queue SYS $PS on host 192.0.0.89.

The **IP SHOW /QUEUE** command is then used to display the contents of both queues; the remote queue first (SYS$PS on FLOWERS) then the local queue (REMOTE_PS).

```
$ PRINT /QUEUE=RE6OTE_PS PROGRAMMERS.PS
Job PROGRAMMERS (queue REMOTE_PS, entry 972) started on REMOTE_PS
$ IP SHOW /QUEUE=REMOTE_PS
Jobname       Username   Entry   Blocks   Status
-------       --------   -----   ------   ------
MANAGE        DAISY      111     988      Printing
```

```
INSTALL       DAISY      115     238      Pending
Printer       queue REMOTE_PS, on FLOWERS::NLP0:"192.0.0.89/SYS$PS"
Jobname       Username   Entry   Blocks   Status
-------       --------   -----   ------   ------
PROGRAMMERS   ROSE       972     1112    Printing at block 370
$
```

This example displays the routing table on the local host without doing IP address-to-name translation.

```
$ IP SHOW /ROUTE /NOSYMB
VSI TCP/IP for OpenVMS IP Routing tables:
Destination      Gateway          Flags          Refcnt   Use   Interface
-----------      -------          _-----------   ------   ---   ---------
192.41.228.129   127.0.0.1        Up,Gateway,H   0        0     lo0
127.0.0.1        127.0.0.1        Up,Host        2        53    lo0
192.41.228.130   192.41.228.129   Up,Host        3        340   sl0
192.41.228.131   192.41.228.129   Up,Host        0        0     dn0
0.0.0            192.41.228.130   Up,Gateway     0        353   sl0
192.41.228.64    192.41.228.65    Up             2        112   se0
192.41.228       192.41.228.1     Up             0        0     pd0
$
```

This example displays local host information.

```
$ IP SHOW/ROUTE/DESTINATIONS=127.0.0.1
VSI TCP/IP for OpenVMS IP Routing tables:
Destination   Gateway     Flags     Refcnt   Use   Interface
-----------   -------     -----     ------   ---   ---------
LOCALHOST     LOCALHOST   Up,Host   1        464   lo0
$
```

This example displays the interface SE1 along with its associated packet filters.

```
$ IP SHOW /INTERFACE SE1 /FILTERS
Device se1: flags=8863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,D2>
  VMS Device = EWB0
  IP Address = 192.168.0.16
  No common links defined
VSI TCP/IP for OpenVMS Packet Filter List for se1:

Logging is disabled

                         Source Address / Port
Action   Proto   Hits    Destination Address / Port
------   -----   -----   ------------------------------------------
deny     tcp     0       192.168.0.11/32
                         192.168.0.0/24  eq 22
                         LOG
         START: 16-MAY-2017 10:33:19  END: 16-MAY-2017 10:38:19
permit   ip      13484   0.0.0.0/0
                         0.0.0.0/0
                      FLTSVR
         Average 0 bytes out, 0 bytes in per second
```

This example displays the current version of VSI TCP/IP and the OpenVMS Operating System.

```
$ IP SHOW /VERSION
VSI TCP/IP for OpenVMS, HP rx2600  (1.30GHz/3.0MB), OpenVMS I64 V8.4-2L1
```

IP TCPDUMP

IP TCPDUMP — Displays the contents of Ethernet packet headers that match the specified boolean expression. To stop the dump, press **Ctrl**/**C**.

Format

`IP TCPDUMP` [*`expression`*]

Restrictions

The following restrictions apply to the use of **IP TCPDUMP**.

- Although the TCPDUMP expression grammar allows the use of the exclamation point (!) character as the NOT operator and as part of the NOT-EQUAL comparator, DCL interprets it as a comment character. Therefore, use NOT instead.

  For example, to print the start and end packets (the SYN and FIN packets) of each TCP conversation that involves a remote host:

  ```
  $ IP TCPDUMP NOT (TCP[13] & 3 = 0) AND NOT SRC -
  _$ AND DST NET LOCALNET
  ```

- PHY_IO, LOG_IO, and SYSPRV or BYPASS privileges are required to use TCPDUMP.
- The packet filter code is not very efficient and adds significant overhead to your OpenVMS system when monitoring a busy network. In addition, if you are using DNS and a problem occurs with name server access, TCPDUMP can appear to hang while waiting for a response from the network.
- IP options are ignored and not displayed.
- Understands PPP frames and does not treat all data as IP datagrams.
- No attempt is made to reassemble IP fragments or at least compute the right length for the higher level protocol.
- Name server inverse queries are not dumped correctly. An empty question section is printed rather than the real query in the answer section.
- Though TCPDUMP recognizes IPsec packets, it does not decrypt encrypted packets.

Parameter

[*`expression`*]

Selects which packets are dumped. If an expression is not given, all packets on the net are dumped. Otherwise, only packets for which the expression is "true" are dumped. Enter HELP IP TCPDUMP EXPRESSION for a list of expression values.

Qualifiers

`/AFTER=`*`time`*

Selects packets dated after the specified time. The time value can be any valid OpenVMS time specification (absolute, delta, or a combination of the two).

`/BEFORE=`*`time`*

Selects packets dated prior to the specified time. The time value can be any valid OpenVMS time specification (absolute, delta, or a combination of the two).

`/COUNT=`*`number_of_packets`*

Exits TCPDUMP after the specified number of packets is received. The default is 0, or no limit.

`/DEBUG`

Displays debugging information.

`/DEVICE=`*`devicename`*

Specifies the OpenVMS device name of the Ethernet device to use. By default, TCPDUMP searches for ECA0, EIA0, EWA0, EZA0, EXA0, EFA0, ETA0, ERA0, ESA0, ICA0, IRA0, LLA0, XEA0,and XQA0 devices.

`/DOMAINS`

`/NODOMAINS`

Displays host names with the domain information; /NODOMAINS strips the domain names.

`/EBCDIC`

Modifies the behavior of the **/HEXADECIMAL** qualifier by adding the EBCDIC translation of the data in addition to the ASCII translation to the TCPDUMP output.

`/ETHERNET_HEADER`

Displays the Ethernet header (source, destination, protocol, and length) on each dump line.

`/FILE_FORMAT=SNIFFER`

Use in conjunction with **/READ_BINARY** or **/WRITE_BINARY** to read or generate output automatically formatted for display on version 2.0 Network General sniffers.

`/FOREIGN_NUMERICALLY`

Displays "foreign" Internet addresses numerically rather than symbolically.

`/HEXADECIMAL_DUMP`

Displays each packet (less its 14-byte Ethernet header) in hexadecimal format. Up to 64 bytes of the packet are printed.

`/INTERFACE=`*`device`*

Specifies the device to trace. Valid devices are those for Ethernet /FDDI (se), the loopback connection (lo0), SLIP lines (sl), PPP lines (ppp), PSI connections (psi), and IP-over-DECNET connections (dn). This qualifier cannot be used with the /DEVICE qualifier.

`/NUMERICALLY`

Specifies that host addresses and port numbers are not converted to names on output.

`/OUTPUT=`*`filename`*

Redirects TCPDUMP output to a file.

`/QUIET`

Specifies that less protocol information is displayed, making output lines shorter.

`/READ_BINARY=`*`binary_file`*

Reads in a file previously written using the **/WRITE_BINARY** qualifier. (Refer to / **WRITE_BINARY** for more information.)

This file is written in libpcap format. When the interface specified is an Ethernet device the data in the file can be analyzed with Ethereal and similar tools.

You can use **/READ_BINARY** with **/FILE_FORMAT=SNIFFER** to read output formatted automatically for display on version 2.0 Network General sniffers. This feature permits sites to analyze Network General analyzer, rather than only examining the TCPDUMP packets. (TCPVIEW also provides the ability to analyze packet traces.)

`/RPC`

Interprets RPC calls in the output.

`/SNAPSHOT_SIZE=`*`snaplen`*

Indicates the specified number of bytes of data to capture from each packet rather than the default of 54 bytes (which is adequate for most applications). 96 bytes is adequate for IP, ICMP, TCP, and UDP, but may truncate protocol information from name server and NFS packets.

`/TIMESTAMPS=`*`value`*

`/NOTIMESTAMPS (default)`

Causes TCPDUMP to display a timestamp on each output line. Accepted values are DEFAULT, UNIX, DELTA, and RELATIVE. The **/NOTIMESTAMPS** qualifier disables the TCPDUMP timestamp on each output line.

`/VERBOSE`

Provides additional information in the output listing.

`/WRITE_BINARY=`*`binary_file`*

Stores the output of TCPDUMP in a file. Use this qualifier to "record" the TCPDUMP information until you press **Ctrl**/**Y**. After recording the output of a TCPDUMP session, use **/READ_BINARY** to read in the binary file for examination.

You can use **/WRITE_BINARY** with **/FILE_FORMAT=SNIFFER** to generate output automatically formatted for display on version 2.0 Network General sniffers. This feature permits sites to analyze Network General analyzer, rather than only examining the TCPDUMP packets. (TCPVIEW also provides the ability to analyze packet traces.)

Examples

This example displays all traffic addressed to or transmitted from host `OL.SLG.COM`.

```
$ IP TCPDUMP HOST OL.SLG.COM
18:56:24.25 BIG.SLG.COM.x11 > OL.SLG.COM.1030:.ack 21527130 win 4096.
```

This example displays all traffic between local hosts and hosts at the network `IRIS-ETHER` at flowers.

```
$ IP TCPDUMP NET IRIS-ETHER
. .
```

This example displays all FTP traffic being sent to host `BETTY.URUB.EDU`.

```
$ IP TCPDUMP - DST HOST BETTY.URUB.EDU AND (PORT FTP OR PORT FTP-DATA)
```

This example displays IP traffic not sent from or destined for the network `IRIS-ETHER`. If `IRIS-ETHER` is the local network, only transient traffic displays.

```
$ IP TCPDUMP IP AND NOT NET IRIS-ETHER
```

IP TCPVIEW

IP TCPVIEW — Traces packets and interprets the results.

Format

`IP TCPVIEW` [*`filename`*]

Parameter

[*`filename`*]

Specifies the name of the optional file to be analyzed. If not specified, TCPVIEW itself can capture network traffic to analyze packets. If you specify a file name, the file must be a Network General Sniffer Version 2 data file, or a TCPDUMP file created with the **TCPDUMP /WRITE_BINARY** qualifier. (Network General data file version IV does not work with TCPVIEW.)

Qualifiers

`/COUNT=`*`number_of_packets`*

If used, TCPVIEW exits after receiving the specified number of packets.

`/DEVICE=`*`devicename`*

Specifies the OpenVMS device name of the Ethernet device to use. By default, TCPVIEW searches for ECA0, EIA0, EWA0, EZA0, EXA0, EFA0, ETA0, ERA0, ESA0, ICA0, IRA0, LLA0, XEA0,and XQA0. This qualifier is provided for backward compatibility; use the **/INTERFACE** qualifier instead. **/DEVICE** bypasses BPF (Berkeley Packet Filter) feature of VSI TCP/IP and allows only a single user to access TCPVIEW. **/DEVICE** cannot be used with the **/INTERFACE** qualifier.

`/DOMAINS (default)`

`/NODOMAINS`

Displays host names with the domain information; **/NODOMAINS** strips the domain names.

`/ETHERNET_HEADER`

Prints the Ethernet header (source, destination, protocol, and length) on each dump line.

`/FILE_FORMAT=SNIFFER`

Specifies that the binary packet trace being read by TCPVIEW is in Network General v2.0 sniffer format.

`/INTERFACE=`*`device`*

Specifies the device to trace. Valid devices are those for Ethernet, FDDI, the loopback connection, SLIP lines, PSI connections, and IP-over-DECnet connections. **/INTERFACE** cannot be used with the **/DEVICE** qualifier.

`/PROMISCUOUS`

Specifies that all network packets are displayed. Using this qualifier adds a significant load to a system.

`/SNAPSHOT_SIZE=`*`snaplen`*

Captures snaplen bytes of data from each packet rather than the default of 54 bytes (which is adequate for many applications). 96 bytes is adequate for IP, ICMP, TCP, and UDP, but may truncate protocol information from name server and NFS packets.

`/TIMESTAMPS`

`/NOTIMESTAMPS (default)`

/NOTIMESTAMPS causes TCPVIEW to suppress printing the timestamp on each output line.

`/VERBOSE`

Causes TCPVIEW to print more verbose packet descriptions.

Description

TCPVIEW provides a mechanism for tracing packets and interpreting the results. The visual interface provides separate windows for the packet trace, an ASCII interpretation of the results, and a hex dump. In addition, as you select interpretation events, the respective sections of the hex dump are highlighted.

TCPVIEW can capture network traffic or read TCPDUMP and Network General Sniffer Version 2 data files. (Version IV does not work with TCPVIEW.) TCPVIEW was derived from TCPDUMP and shares many characteristics with it. It must be run from a privileged account, but should not be installed with privileges. TCPVIEW uses DECwindows/Motif, which must be installed along with DECwindows to permit TCPVIEW to run. (TCPVIEW will not run with DECwindows **alone**.)

The main display is a window with three resizeable panes.

- The top pane contains a summary line describing each packet. This line is identical to the output of TCPDUMP. Selecting a line in the top pane activates the middle and bottom panes.

- The middle pane contains a detailed decoding of the selected frame. Information is only included here if the appropriate protocol decoders are present. If a line is selected in this pane, the corresponding line will be at the top of this pane for all subsequent frames decoded.
- The bottom pane is a hex dump of the entire frame. Data is highlighted when a line is selected in the middle pane.

TCPVIEW MENUS

The TCPVIEW menu bar allows you to change configuration settings, load and store files, access help, and exit. The menu bar consists of File, Capture, Filter, Options, and Help pulldown entries.

FILE MENU

The File pulldown menu allows you to open (load) a data file, save (store) a data file, print a packet trace, and exit TCPVIEW.

- File Open displays a DECwindows/Motif file dialog box that lists the files in the current directory. To load a file, click the required file, then click OK. (Double-clicking selects and loads the file in one operation.)
- File Save displays the Save dialog box which allows you to save only the filtered packets or all of the seen packets. This selection also allows you to determine the format of the output file. A text field is provided to assign a file name to the output file.
- File Print displays the Print dialog box which allows you to print all or only filtered packets in either Summary or Detail mode.
- Summary mode displays a single line, much like the standard TCPDUMP output.
- Detail mode breaks each packet down, much like the Network General Sniffer output.
- File Exit allows you to exit the TCPVIEW utility.

CAPTURE MENU

The Capture pulldown menu sets capture options and begins capturing packets.

- Capture Set Options specifies the parameters that control how a packet is captured. This selection contains these options:
  - Device Name selects which device interface (VSI TCP/IP) to use for capturing data.
  - Promiscuous Mode determines if the interface is set to promiscuous mode. If promiscuous mode is not enabled, you can only capture packets using the VSI TCP/IP interface (which supports the BPF packet interface).
  - Number of Frames sets a limit on the number of frames that can be captured. Numbers less than or equal to 0 and invalid entries reset the limit to "infinite".
  - Time Limit sets a limit on the number of seconds that data will be captured. Numbers less than or equal to 0 and invalid entries reset the limit to "infinite".
  - Max Bytes Per Frame sets the maximum number of bytes that can be captured per frame (the minimum is 68 bytes); sizes smaller than the minimum are not accepted.

- Capture GO starts the capture of frames. Stop a capture as follows:

  1. When the Stop button appears, click it or press **RETURN**.
  2. Wait until the maximum time is reached, or until the maximum packets to be captured is reached.

FILTER MENU

The Filter pulldown menu allows you to edit the expression that controls the frame filter. A frame filter is required for a given capture.

The Filter Edit option allows you to set up an expression used to filter the captured frames. (This is similar to the capabilities provided by the TCPDUMP expression syntax.)

There are two address filters. To activate one, click the **OFF** button. If both filters are activated, the second line toggle button switches to **AND**. Click it again to change it to **OR**.

The filters can work on either Data Link Level (DLC) or IP addresses. To change the address:

1. Click the **ANY** button. A request box appears asking for the new DLC or IP address.
2. Use the address filter to select the DLC or IP address to apply to the current data or the data to be captured.
3. Click any of the buttons to either toggle the button's state or display a request box for new information.

Enter **ANY** or **ALL** (case-insensitive) to set a filter back to the **ANY** state. For a numeric Ethernet address, enter the address in hex format either starting with "0x" or as six bytes separated by colons (for example, 0x08202b000002 or 08:20:2B:00:00:02). For IP addresses, enter a name or numeric address such as 161.44.128.70.

- The Protocol filter allows you to select the protocols you want to capture. You can select all to see any protocol you want, or select one or more of the protocols provided.
- The Port filter allows you to select all packets with that port as a source or destination. You can enter either a port number or a name. Port names are assigned via `IP$:HOSTS.SERVICES` and `IP$:HOSTS.LOCAL`. If the port name cannot be found, the filter is reset to ANY.
- The Clear filter button resets the filter to its initial state (allow all packets).
- The Apply To All applies the filter to all data that is currently captured. Selecting this with no filter in place displays all captured frames.
- Apply To Current applies the filter only to the current selected list of frames, allowing you to re-filter a captured set of packets repeatedly to look for problems or something specific in the data.

OPTIONS MENU

The Options pulldown menu controls how the data is presented, and contains options to control the address, time, and miscellaneous options.

- Address Options allow you to control how the address is presented in the main window. You can display the host name, IP address, or DLC. If you select the host name, you can choose that the Fully Qualified Domain Name (FQDN) is displayed; otherwise, the short name is used.

If you are displaying the DLC, you can use the manufacturer's names instead of the DLC. This information is provided in the file `IP$:MANUF.CODES`. For example, use this option to display PSC_003462 instead of 00000C003462 (00000C is the assigned Ethernet code for VSI).

- Time Options control how the timestamp associated with a packet is displayed.
- Absolute prints the arrival time in the format "HH:MM:SS.SS".
- UNIX displays timestamps in the UNIX format, which is the number of seconds since 00:00:00 GMT (January 1, 1970).
- Delta prints the timestamp as the number of elapsed seconds between frames.
- Relative prints the number of seconds from the first frame.
- None does not print a timestamp.
- Miscellaneous Options specify how each frame is displayed.
- Verbose provides additional information, such as displaying the time-to-live (TTL) and the type of service information in an IP packet.
- Brief displays a minimum amount of protocol information.
- Display DLC header displays the DLC source, destination, and protocol type in the summary line.
- Use Relative TCP Sequence Numbers resets each TCP connection's sequence to make it easier to follow.
- Display Line Numbers displays a number for each frame.

HELP MENU

The Help pulldown menu provides access to online help. It provides three options in the pulldown: Overview, About, and Help On Help. It can also provide context-sensitive help by using MB1 and the Help button. For help on context-sensitive help, see the help provided by this widget or see the *DECwindows/Motif User's Guide*.

IP TRACEROUTE

IP TRACEROUTE — Attempts to trace the route that an IP packet follows to another Internet host.

Format

`IP TRACEROUTE` *`host`* [*`data_length`*]

Description

TRACEROUTE /IPV6 finds the intermediate hops by sending probe packets with a small TTL (time-to-live), then listening for an ICMP "time exceeded" reply from a gateway. It starts probing with a TTL of one, then increases by one in each successive probe until an ICMP "port unreachable" reply is received (indicating that a probe reached the host) or the TTL exceeded 30 (the default maximum).

By default, three probes are sent at each TTL setting, and a line is printed showing the TTL, the gateway address, and round trip time of each probe. If the probe answers come from different gateways, the address of each responding system is printed. If there is no response within a five-second timeout interval, a "*" is printed for that probe. **TRACEROUTE** prints a "!" after the time if the TTL is less than or equal to one. The following table shows other possible annotations:

Annotation	Meaning
`!H`	Host unreachable
`!N`	Network unreachable
`!P`	Protocol unreachable
`!S`	Source route failed
`!F`	Fragmentation_needed

The **!S** and **!F** annotations are rare and indicate that the associated gateway is not working properly. If most of the probes result in "unreachable" annotations, **TRACEROUTE** stops running and exits.

TRACEROUTE /IPV6 performs the same functions as **TRACEROUTE** for IPv6.

Parameters

`host`

Specifies the target host to which you want to determine the route.

[*`data_length`*]

Specifies the amount of data sent in each ICMP Echo Request packet.

Qualifiers

`/DEBUG`

`/NODEBUG (default)`

Enables socket-level debugging in the VSI TCP/IP kernel. This qualifier is used only for debugging the VSI TCP/IP kernel.

`/IPV6`

Specifies that an IPv6 trace is desired. (The default is IPv4.)

`/MAXIMUM_TTL=`*`maximum_ttl`*

Specifies the maximum TTL (time-to-live) to explore looking for ICMP Time Exceeded responses. If not specified, the default of 30 hops is used.

`/MINIMUM_TTL=`*`minimum_ttl`*

Specifies the minimum TTL to explore looking for ICMP Time Exceeded responses. If not specified, the default of 1 hop is used.

`/NUMBER_OF_PROBES=`*`n`*

Specifies the number of probe packets sent to each hop (by default, 3).

`/OUTPUT=`*`filename`*

Redirects TRACEROUTE output to a file.

/PORT=*udp_port*

Specifies a non-standard port number. **TRACEROUTE** sends data to an unused port and expects an error message. If the default port of 33434 is in use, use **/PORT** to specify another.

/ROUTE (default)

/NOROUTE

Disables any IP routing of the ICMP packets. The default, **/ROUTE**, allows IP routing to send the packet to destinations separated by gateways.

/SOURCE=*ip_address*

Specifies the local IP address from which packets are sent.

/SYMBOLIC_ADDRESSES (default)

/NOSYMBOLIC_ADDRESSES

Specifies that IP addresses are displayed numerically instead of being converted into host names.

/TYPE_OF_SERVICE=*tos*

Specifies the Type-Of-Service (TOS) field of the IP packet. The default TOS is 0 (no specific type of service).

/VERBOSE

/NOVERBOSE (default)

Displays extra information as ICMP packets are sent or received.

/WAIT_TIME=*seconds*

Specifies how long **TRACEROUTE** waits for responses (by default, 5 seconds).

Examples

This example shows tracing a route to an NSFnet gateway. Note: lines 2 and 3 are the same. This is because the gateway "lilac-dmc.Berkeley.Edu" has a kernel bug that causes the system to forward packets with a TTL of zero.

```
$ IP TRACEROUTE NIS.NSF.NET
traceroute to nis.nsf.net (35.1.1.48), 30 hops max, 38 byte packet
1   FLOWERS.BARRNET.NET (192.41.228.71)       0 ms   0 ms   0 ms
2   UCSC.BARRNET.NET (131.119.46.7)           10 ms  0 ms   20 ms
3   SU1.BARRNET.NET (131.119.1.5)             10 ms  20 ms  20 ms
4   SU-B.BARRNET.NET (131.119.254.201)        20 ms  20 ms  20 ms
5   E-NSS.BARRNET.NET (192.31.48.244)         50 ms  10 ms  20 ms
6   t3-1.cnss9.t3.nsf.net (140.222.9.2)       20 ms  10 ms  20 ms
7   t3-3.cnss8.t3.nsf.net (140.222.8.4)       20 ms  30 ms  30 ms
8   t3-0.cnss24.t3.nsf.net (140.222.24.1)     70 ms  60 ms  60 ms
9   t3-0.cnss40.t3.nsf.net (140.222.40.1)     70 ms  70 ms  60 ms
10  t3-0.cnss41.t3.nsf.net (140.222.41.1)     70 ms  70 ms  60 ms
```

```
11  t3-0.enss131.t3.nsf.net (140.222.131.1)   70 ms  80 ms  80 ms
12  nis.nsf.net (35.1.1.48)                   80 ms  80 ms  70 ms
$
```

IP X11DEBUG

IP X11DEBUG — The **X11DEBUG** utility performs four tests that check the most common causes of problems encountered when running X11 clients over VSI TCP/IP.

Format

```
IP X11DEBUG
```

Description

The most common causes of problems when running X11 clients over VSI TCP/IP are:

- Checks for the UCX driver.
- Verifies that a DISPLAY has been defined with the **SET DISPLAY** command.
- Checks TCP/IP connections.
- Verifies that the X11 client can access the server.

If any of these tests fail, **IP X11DEBUG** recommends a course of action to resolve the problem. Otherwise, **IP X11DEBUG** displays the message, "%X11DEBUG-S-PASSEDALL, passed all X11 tests."

Qualifier

`/LOG`

`/NOLOG (default)`

Enables additional debugging information.

Chapter 2. DECNET-CONFIG Command Reference

This chapter describes the commands you can run from the DECNET-CONFIG command line. DECNET-CONFIG lets you examine, modify, and save configuration files for DECnet-over-IP circuits.

To invoke DECNET-CONFIG:

```
$ IP CONFIGURE /DECNET
```

At any DECNET-CONFIG prompt, type ? to list the available commands. Use the command to view online help for each DECNET-CONFIG command.

Changes do not take effect until you do one of the following:

- Restart the DECnet-over-IP driver with the command

  ```
  @IP$:DECNET-CIRCUITS.COM.
  ```

- Restart your system.

For details on configuring DECnet-over-IP circuits, refer to the *VSI TCP/IP Administrator's Guide: Volume II.*

Command Summary

Table 2.1 lists the commands you can use from the DECNET-CONFIG prompt.

Table 2.1. DECNET-CONFIG Command Summary

Command	Description
ADD	Adds a DECnet-over-IP circuit to the current configuration.
ATTACH	Attaches to a subordinate process.
CLEAR	Deletes all DECnet-over-IP circuits from the current configuration; same as **ERASE**.
DELETE	Deletes a single DECnet-over-IP circuit from the current configuration.
ERASE	Deletes all DECnet-over-IP circuits from the current configuration; same as **CLEAR**.
EXIT	Exits DECNET-CONFIG and saves the current configuration.
GET	Reads in a DECnet-over-IP circuit configuration file; same as **USE**.
HELP	Displays command information.
MODIFY	Changes the parameters in a DECnet-over-IP circuit configuration file.
PUSH	Accesses the DCL command line and pauses DECNET-CONFIG.

Command	**Description**
QUIT	Exits DECNET-CONFIG, but prompts to save changes before exiting.
SAVE	Writes out the DECNET-CONFIG current configuration file; same as **WRITE**.
SHOW	Displays the current DECnet-over-IP circuit configuration.
SPAWN	Executes a single DCL command or starts a subprocess.
STATUS	Displays the status of the DECnet-over-IP circuit configuration.
USE	Reads in a DECnet-over-IP circuit configuration file; same as **GET**.
VERSION	Displays DECNET-CONFIG version and release information.
WRITE	Writes out the current DECNET-CONFIG configuration file; same as **SAVE**.

ADD

ADD — Adds a new DECnet-over-IP circuit to the current VSI TCP/IP configuration, and prompts for circuit configuration parameters.

Format

`ADD` [*`circuit_name`*]

Parameter

[*`circuit_name`*]

Specifies the name of the DECnet circuit to add to the configuration. If not specified in the command, the first unused circuit in the configuration is used by default.

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
DECNET-CONFIG>ADD
[Adding new configuration entry for DECnet circuit "TCP-0-0"]
Destination IP Address: [NONE] 192.0.0.6
DECnet circuit cost: [1] 1
DECnet hello timer (in seconds): [300] 300
[TCP-0-0 => 192.0.0.6 (Cost=1, Hello Timer=300)
DECNET-CONFIG>EXIT
[Writing configuration to IP$:DECNET-CIRCUITS.COM]
$
```

ATTACH

ATTACH — Detaches the terminal from the calling process and reattaches it to another process. Use the command to list the names of the subprocesses. Use the DCL command to return to the original process. If the `IP$DISABLE_SPAWN` logical is enabled, does not work.

Format

```
ATTACH process-name
```

Parameter

process-name

Specifies the name of a process to which you want your terminal attached. (Not all subprocesses can be attached; some testing may be required.)

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM
DECNET-CONFIG>SPAWN
$ MM
MM>SPAWN SHOW PROCESS/SUB
...
There are 3 processes in this job:
_TWA42:
PROC_1
PROC_2 (*)
MM>ATTACH _TWA42:
DECNET-CONFIG>ATTACH PROC_1
MM>QUIT
$ LOGOUT
DECNET-CONFIG>
```

This example shows the use and exit of attached subprocesses.

1. The first command uses SPAWN to create a subprocess. MM is invoked from the DCL command line. Next, the command is used to list all subprocess names. The display shows that three subprocesses are active. (Process _TWA42: is DECNET-CONFIG, PROC_1 is MM, and PROC_2 is the command.)

2. In the next command, the command returns control to the DECNET-CONFIG process. From this utility, returns control to MM. To exit, the command is invoked from , and is invoked at the original spawned DCL command line; control returns to DECNET-CONFIG. (If **SPAWN SHOW PROCESS /SUB** had been entered, only this command and the configuration processes would be active.)

CLEAR

CLEAR — Deletes all DECnet-over-IP circuits from the current VSI TCP/IP configuration.

Format

```
CLEAR
```

Example

```
$ IP CONFIGURE /DECNET
```

```
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>CLEAR
DECNET-CONFIG>EXIT
$
```

DELETE

DELETE — Deletes the specified DECnet-over-IP circuit from the current VSI TCP/IP configuration.

Format

DELETE *circuit_name*

Parameter

circuit_name

Specifies the name of the DECnet-over-IP circuit to delete.

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>DELETE TCP-0-0
DECNET-CONFIG>EXIT
$
```

ERASE

ERASE — Clears all DECnet-over-IP circuits from the current VSI TCP/IP configuration. (Functionally equivalent to **CLEAR**.)

Format

ERASE

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>ERASE
DECNET-CONFIG>EXIT
$
```

EXIT

EXIT — Saves the current configuration, if it has been modified, then quits.

Format

```
EXIT
```

Examples

If the configuration has not changed, a message displays indicating that the configuration file is not updated.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>EXIT
$
```

If the configuration has changed, a message displays indicating that the configuration file has been updated.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>CLEAR
DECNET-CONFIG>EXIT
[Writing configuration to IP$:DECNET-CIRCUITS.COM}
$
```

GET

GET — Reads in a DECnet-over-IP circuit configuration file, which defaults to the highest version number of the `IP$:DECNET-CIRCUITS.COM` file. (Functionally equivalent to **USE**.) After using , you can use other DECNET-CONFIG commands to display and modify the new configuration.

Format

GET *config_file*

Parameter

config_file

Specifies the name of the configuration file to read in.

Example

This example reads in the highest version number of the `IP$ROOT:[IP]DECNET-CIRCUITS.COM` configuration file. In this case, the highest version number is ";7". This number is appended to the file name as the value ".7".

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>GET
[Reading in configuration from IP$ROOT:[IP]DECNET-CIRCUITS.COM.7]
DECNET-CONFIG>EXIT
```

```
$
```

HELP

HELP — Invokes the command.

Format

`HELP` [*topics*]

Parameter

[*topics*]

Contains a space-delimited list of topics that begins with a topic followed by subtopics. The default topic is .

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>HELP ?
ADD       ATTACH   CLEAR   DELETE   ERASE   EXIT    GENERAL   GET
MODIFY    PUSH     QUIT    SAVE     SHOW    SPAWN   STATUS    USE
VERSION   WRITE
DECNET-CONFIG>
```

MODIFY

MODIFY — Changes the parameters of the specified DECnet-over-IP Circuit in the VSI TCP/IP configuration.

Format

`MODIFY` *circuit_name*

Parameter

circuit_name

Specifies the name of the DECnet-over-IP circuit whose parameters will be modified.

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
DECNET-CONFIG>MODIFY TCP-0-0
[Adding new configuration entry for DECnet circuit “TCP-0-0”]
Destination IP Address: [192.0.0.6]
DECnet circuit cost: [1] 5
DECnet hello timer (in seconds): [300]
```

```
[TCP-0-0 => 192.0.0.6 (Cost=5, Hello Timer=300)]
DECNET-CONFIG>EXIT
$
```

PUSH

PUSH — Starts and attaches a DCL subprocess. If a parent process exists, attach to it. To return from DCL, use the or the command. To switch back from a DCL subprocess, use the command. If the `IP $DISABLE_SPAWN` logical is set, does not work.

Format

```
PUSH
```

Example

In this example, is used to access the DCL command line to disable broadcasts. The command returns control to DECNET-CONFIG.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>PUSH
$ SET TERM /NOBROADCAST
$ LOGOUT
DECNET-CONFIG>
```

QUIT

QUIT — If the configuration file has been edited, prompts you to save the file before quitting.

Format

```
QUIT
```

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>QUIT
Configuration modified, do you want to save it ? [NO]NO
```

SAVE

SAVE — Writes out the current configuration file. (Functionally equivalent to **WRITE**; see for additional information.)

Format

SAVE

SHOW

SHOW — Displays the current VSI TCP/IP DECnet-over-IP circuit configuration.

Format

`SHOW`

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>SHOW
Circuit Name   IP Destination   Cost   Hello Timer
------------   --------------   ----   -----------
TCP-0-0        192.26.30.1      1      300
DECNET-CONFIG>QUIT
```

SPAWN

SPAWN — Executes a single DCL command, or if entered without options, starts a subprocess with the same effect as . To return from DCL, use the command. If the `IP$DISABLE_SPAWN` logical is set, does not work.

Format

`SPAWN` *`command`*

Parameter

`command`

Specifies a command to execute. If you omit *`command`*, a DCL command line subprocess is created.

Qualifiers

`/INPUT=`*`file-spec`*

Specifies an input file to the command you enter with .

`/LOGICAL_NAMES`

`/NOLOGICAL_NAMES`

Specifies that logical names and logical name tables are not copied to the subprocess.

`/SYMBOLS`

`/NOSYMBOLS`

Specifies that global and local names are not passed to the subprocess.

`/WAIT`

/NOWAIT

Returns control without waiting for the command to complete. Do not use this qualifier with commands that have prompts or screen displays.

/OUTPUT=*file-spec*

Specifies a file that retains the output of the command invoked with . This qualifier only works when a single command is entered without creating a DCL subprocess. In addition, this qualifier is positional; you must enter it immediately after or other qualifiers.

Examples

This example displays terminal information, captures the output in a file, and displays the information with the command.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>SPAWN/OUTPUT=FOO. SHOW TERM
DECNET-CONFIG>SPAWN TYPE FOO.
...
```

This example invokes a command procedure.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>SPAWN @COMPROC
...
```

This example displays help information about DECNET-CONFIG. Use the command to return control to DECNET-CONFIG.

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>SPAWN
$ HELP IP CONFIGURE /DECNET ...
$ LOGOUT
DECNET-CONFIG>
```

STATUS

STATUS — Displays the status of the VSI TCP/IP DECnet-over-IP circuit configuration.

Format

```
STATUS
```

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
```

```
DECNET-CONFIG>STATUS
This is the VSI TCP/IP for OpenVMS DECnet circuit configuration program
 Version 10.5 (nnn)
There are 1/100 circuits in the current configuration.
The configuration IP$:DECNET-CIRCUITS.COM is not modified.
DECNET-CONFIG>QUIT
$
```

USE

USE — Reads in a configuration file. (Functionally equivalent to **GET**; see **GET** for additional information.)

Format

`USE` *`config_file`*

Parameter

`config_file`

Specifies the name of the configuration file to read in.

VERSION

VERSION — Displays the version and release information of the VSI TCP/IP DECnet-over-IP circuit configuration program.

Format

`VERSION`

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>VERSION
This is the VSI TCP/IP for OpenVMS DECnet circuit configuration program
 Version 10.5 (nnn)
DECNET-CONFIG>QUIT
$
```

WRITE

WRITE — Writes the current VSI TCP/IP DECnet-over-IP circuit configuration to a file. (Functionally equivalent to **SAVE**.)

Format

`WRITE` [*`config_file`*]

Parameter

[*config_file*]

Specifies the name of the DECnet-over-IP circuit configuration file to write out (by default, the same file from which the configuration was read).

Example

```
$ IP CONFIGURE /DECNET
VSI TCP/IP for OpenVMS DECNET Circuit Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:DECNET-CIRCUITS.COM]
DECNET-CONFIG>WRITE
[Writing configuration to IP$ROOT:[IP]DECNET CIRCUITS.COM.2]
DECNET-CONFIG>EXIT
```

Chapter 3. MAIL-CONFIG Command Reference

This chapter describes the commands you can run from the MAIL-CONFIG command line. MAIL-CONFIG lets you examine, modify, and save configuration files for the VSI TCP/IP SMTP mail system.

To invoke MAIL-CONFIG:

```
$ IP CONFIGURE /MAIL
```

At any MAIL-CONFIG prompt, type ? to list the available commands. Use the MAIL-CONFIG **HELP** command to view online help for each MAIL-CONFIG command.

Changes do not take effect until you do one of the following:

- Eanble SMTP as a service with the following command:

  ```
  $ IP CONFIGURE/SERVER
  VSI TCP/IP Server Configuration Utility
  [Reading in configuration from IP$:SERVICES.MASTER_SERVER]
  SERVER-CONFIG>SELECT SMTP
  [The Selected SERVER entry is now SMTP]
  SERVER-CONFIG>ENABLE SMTP
  SERVER-CONFIG>EXIT
  ```

- Restart the SMTP service with the commands:

  ```
  @IP$:START_SMTP.COM
  ```

  or

  ```
  @IP$:START_SMTP_LOCAL.COM
  ```

- Restart your system.

For details on configuring electronic mail, refer to the *VSI TCP/IP Administrator's Guide: Volume I.*

Command Summary

Table 3.1 lists the commands you can run from the MAIL-CONFIG prompt.

Table 3.1. MAIL-CONFIG Command Summary

MAIL-CONFIG Command	Description
ADD GATEWAY	Adds a mail gateway to another domain.
ADD LOCAL-DOMAIN	Adds a domain to a list of domains that the VSI TCP/IP SMTP symbiont considers to be local. If users send mail to hosts beyond the local domains, VSI TCP/IP forwards the mail to the mail hub specified by the *`FORWARDER`* parameter. The local domain list affects mail forwarding only when the *`FORWARD-REMOTE-MAIL`* parameter is TRUE.
ADD QUEUE-GROUP	Forms a mail queue grouping of nodes in a cluster, or adds new nodes to an existing queue group.

MAIL-CONFIG Command	Description
ATTACH	Attaches your terminal to another process.
CLEAR	Erases all information from the current configuration; same as **ERASE**.
DELETE GATEWAY	Deletes a mail gateway.
DELETE LOCAL-DOMAIN	Deletes a domain from VSI TCP/IP list of local domains.
DELETE QUEUE-GROUP	Deletes a queue group or removes a node from a queue group. When a node is removed from a named queue group, it becomes part of the default queue group.
ERASE	Erases all information from the current configuration; same as **CLEAR**.
EXIT	Saves the configuration file and exits from MAIL-CONFIG.
GET	Reads in a VSI TCP/IP SMTP configuration file. (Functionally equivalent to **USE**.)
HELP	Invokes MAIL-CONFIG command help.
PUSH	Accesses the DCL command interpreter.
QUIT	Prompts you to save the configuration file if it has been modified, then exits MAIL-CONFIG.
REMOVE GATEWAY	Functionally equivalent to **DELETE GATEWAY**.
REMOVE QUEUE-GROUP	Functionally equivalent to **DELETE QUEUE-GROUP**.
SAVE	Saves the current configuration file.
SET ACCOUNTING-HOST	Sets the host that SMTP accounting information should be sent to.
SET ACCOUNTING-PORT	Sets the port that the accounting logger is to listen on.
SET ALIAS-FILE	Identifies the file that holds mail aliases.
SET DECNET-DOMAIN	Sets the domain name for DECnet mail.
SET DELIVERY-RECEIPTS	Specifies whether mail receipts are sent when incoming mail containing Delivery-Receipt-To: or Return-Receipt-To: headers is submitted to the SMTP queue.
SET DISALLOW-USER-REPLY-TO	When set to TRUE, prevents OpenVMS MAIL users from setting a Reply-To: header address with the logical name `IP $SMTP_REPLY_TO`.
SET FORWARDER	Specifies the host that will forward mail messages to other hosts.
SET FORWARD-LOCAL-MAIL	Forwards mail addressed to users on the local host to a central mail hub specified by the *`FORWARDER`* parameter.
SET FORWARD-REMOTE-MAIL	Forwards mail addressed to users on non-local hosts to a central mail hub specified by the *`FORWARDER`* parameter.
SET HEADER-CONTROL	Specifies which RFC-822 message headers should be included in messages delivered to local OpenVMS MAIL users.
SET HOST-ALIAS-FILE	Specifies a file from which VSI TCP/IP obtains a list of hosts aliases.
SET LOCAL-MAIL-FORWARDER	Forwards local mail to a specific host.
SET POSTMASTER	Identifies the user responsible for mail on the system.

MAIL-CONFIG Command	Description
SET QUEUE-COUNT	Specifies the number of mail processing queues that should be created on a particular system.
SET REPLY-CONTROL	Specifies how Internet mail headers should be mapped to the OpenVMS MAIL "From" header.
SET RESENT-HEADERS	When FALSE, the VSI TCP/IP SMTP symbiont omits the Resent-From, Resent-To, and Resent-Date headers that are usually included when a message is forwarded using a OpenVMS MAIL forwarding address.
SET RETRY-INTERVAL	Specifies the amount of time that elapses before another attempt is made to send a message after a failed attempt.
SET RETURN-INTERVAL	Specifies the amount of time that a message can remain in the processing queue before it is returned to sender.
SET SEND-BROADCAST-CLASS	Specifies the broadcast class to use to deliver immediate SEND messages.
SET SMTP-HOST-NAMES	Sets the host name from which all outgoing mail appears to be sent and aliases for which this host accepts incoming mail.
SET START-QUEUE-MANAGER	Determines whether `START_SMTP.COM` starts the OpenVMS queue manager if it is not already running.
SHOW	Displays the current configuration.
SPAWN	Executes a single DCL command.
STATUS	Indicates whether the SMTP configuration has been modified.
USE	Reads in a non-standard configuration file.
VERSION	Displays the MAIL-CONFIG version and release information.
WRITE	Saves the current configuration file.

ADD GATEWAY

ADD GATEWAY — Adds a mail gateway to another domain. Specifies a gateway host to which mail for the specified host or domain will be forwarded. To define a mail gateway to an IP address (instead of a host name), you must enclose the IP address in square brackets. ***Note: In VSI TCP/IP, only one gateway is allowed to be defined per domain. Preference numbers are not allowed.***

Format

`ADD GATEWAY` *`domain_name hostname`*

Parameters

`domain_name`

Specifies the name of the domain for which the new gateway will handle mail. This can be a fully qualified host name (for example, `WHORFIN.FLOWERS.COM`) or a domain tag beginning with a dot (for example, .BITNET).

`hostname` `or` *`nn.nn.nn.nn`*

Specifies the name of the host that acts as a gateway for mail addressed to domain_name.

ADD LOCAL-DOMAIN

ADD LOCAL-DOMAIN — Adds a domain to a list of domains that the VSI TCP/IP SMTP symbiont considers to be local. If users send mail to hosts beyond the local domains, VSI TCP/IP forwards the mail to the mail hub specified by the *`FORWARDER`* parameter. The local domain list affects mail forwarding only when the *`FORWARD-REMOTE-MAIL`* parameter is TRUE.

Format

`ADD LOCAL-DOMAIN` *`domain_name`*

Parameter

`domain_name`

Specifies the name of a domain (for example, `LOT-49.FLOWERS.COM`) that VSI TCP/IP considers to be local.

ADD QUEUE-GROUP

ADD QUEUE-GROUP — Forms a mail queue grouping of nodes in a cluster, or adds new nodes to an existing queue group. The SMTP queues on the nodes in the group you create will share responsibility for handling mail messages generated on nodes within the group. If a node is not placed in a named queue group, it is made part of the default queue group.

Format

`ADD QUEUE-GROUP` *`groupname`* [*`node_name_list`*]

Parameters

`groupname`

Specifies the name of the queue group to add, or the name of an existing group to which nodes will be added.

[*`node_name_list`*]

Contains a list of names of OpenVMScluster (SCS) nodes to add to the queue group.

ATTACH

ATTACH — Detaches the terminal from the calling process and reattaches it to another process. Use the SPAWN **SHOW PROCESS /SUBPROCESSES** command to list the names of the subprocesses. Use the DCL **LOGOUT** command to return to the original process. If the `IP$DISABLE_SPAWN` logical is enabled, **ATTACH** does not work.

Format

`ATTACH` *`process-name`*

Parameter

`process-name`

Specifies the name of a process to which you want your terminal attached. (Not all subprocesses can be attached; some testing may be required.)

CLEAR

CLEAR — Clears all information from the current configuration. (Functionally equivalent to **ERASE**.)

Format

```
CLEAR
```

DELETE GATEWAY

DELETE GATEWAY — Deletes a mail gateway.

Format

`DELETE GATEWAY` *`domain_name`*

Parameter

`domain_name`

Specifies the name of the domain whose gateway will be deleted.

DELETE LOCAL-DOMAIN

DELETE LOCAL-DOMAIN — Deletes a domain from VSI TCP/IP list of local domains.

Format

`DELETE LOCAL-DOMAIN` *`domain_name`*

Parameters

`domain_name`

Specifies the name of the domain to delete from the list of local domains.

DELETE QUEUE-GROUP

DELETE QUEUE-GROUP — Deletes a queue group or removes a node from a queue group. When a node is removed from a named queue group, it becomes part of the default queue group.

Format

`DELETE QUEUE-GROUP` *`group_name`* [*`node_names`*]

Parameters

`group_name`

Specifies the name of the group to delete or the name of the group from which to remove the specified nodes.

[*`node_names`*]

Specifies the OpenVMScluster (SCS) node name to remove from the specified queue group.

ERASE

ERASE — Erases all information from the current configuration. (Functionally equivalent to **CLEAR**.)

Format

```
ERASE
```

EXIT

EXIT — Saves the current configuration, if it has been modified, then quits.

Format

```
EXIT
```

GET

GET — Reads in a VSI TCP/IP SMTP configuration file. (Functionally equivalent to **USE**.) After a **GET**, you can use the various configuration commands to modify the SMTP configuration.

Format

`GET` *`config_file`*

Parameter

`config_file`

Specifies the name of the SMTP configuration file to read in.

HELP

HELP — Invokes command help.

Format

`HELP` *`topics`*

Parameter

`topics`

Contains a space-delimited list of topics that begins with a topic followed by subtopics. The default topic is **HELP**.

PUSH

PUSH — Starts and attaches a DCL subprocess. If a parent process exists, attach to it. To return from DCL, use the **ATTACH** or the **LOGOUT** command. To switch back from a DCL subprocess, use the **ATTACH** command. If the `IP$DISABLE_SPAWN` logical is set, PUSH does not work.

Format

```
PUSH
```

QUIT

QUIT — If the configuration file has been edited, **QUIT** prompts you to save the file before quitting.

Format

```
QUIT
```

REMOVE GATEWAY

REMOVE GATEWAY — Functionally equivalent to **DELETE GATEWAY**.

Format

`REMOVE GATEWAY` *`domain_name`*

Parameter

`domain_name`

Specifies the name of the gateway to remove.

REMOVE QUEUE-GROUP

REMOVE QUEUE-GROUP — Functionally equivalent to **DELETE QUEUE-GROUP**.

Format

`REMOVE QUEUE-GROUP` *`group_name`* [*`node_names`*]

Parameters

`group_name`

Specifies the name of the group to remove or the name of the group from which to remove the specified nodes.

[*`node_names`*]

Specifies the OpenVMScluster (SCS) node name to remove from the specified queue group.

SAVE

SAVE — Writes the current VSI TCP/IP SMTP configuration to SMTP configuration files. (Functionally equivalent to **WRITE**.)

Format

`SAVE` *`config_file`*

Parameter

`config_file`

Specifies the name of the file to which to write the current VSI TCP/IP SMTP configuration (by default, the same file from which it was read).

SET ACCOUNTING-HOST

SET ACCOUNTING-HOST — Sets the host that SMTP accounting information should be sent to.

Format

`SET ACCOUNTING-HOST` *`hostname`*

Parameter

`hostname`

Specifies the name of the host that SMTP accounting information should be sent to. Accounting-Port must be specified also.

SET ACCOUNTING-PORT

SET ACCOUNTING-PORT — Sets the port that the accounting logger is to listen on.

Format

`SET ACCOUNTING-PORT` *`port number`*

Parameter

`port number`

Specifies the port number that the accounting logger is listening on. Accounting-Host must be specified also.

SET ALIAS-FILE

SET ALIAS-FILE — Identifies the file that holds system-wide mail aliases.

Format

`SET ALIAS-FILE`[*`file-spec`*]

Parameter

[*`file-spec`*]

Specifies the name of the file that contains system-wide mail aliases (by default, `IP$:SMTP_ALIASES`).

SET DECNET-DOMAIN

SET DECNET-DOMAIN — Sets the domain name for DECnet mail.

Format

`SET DECNET-DOMAIN` *`domain_name`*

Parameter

`domain_name`

Specifies the domain name for DECnet mail. For information on the SMTP-DECnet gateway, see the *VSI TCP/IP Administrator's Guide: Volume I.*

SET DELIVERY-RECEIPTS

SET DELIVERY-RECEIPTS — Specifies whether mail receipts are sent when incoming mail containing Delivery-Receipt-To: or Return-Receipt-To: headers is submitted to the SMTP queue. If TRUE, mail receipts are sent.

Format

`SET DELIVERY-RECEIPTS` { TRUE | FALSE }

SET DISALLOW-USER-REPLY-TO

SET DISALLOW-USER-REPLY-TO — When set to TRUE, prevents OpenVMS MAIL users from setting a Reply-To: header address with the `IP$SMTP_REPLY_TO` logical name.

Format

`SET DISALLOW-USER-REPLY-TO` { TRUE | FALSE }

SET FORWARDER

SET FORWARDER — Specifies the host that will forward mail messages to other hosts.

Format

`SET FORWARDER` [*`host_name`*]

Parameter

[*`host_name`*]

Specifies the name of the host to which mail is forwarded when attempts by the local system to send mail to a remote system fail because of a host name lookup failure.

If no host name is specified, no forwarder is used, and failed messages are tried repeatedly (based on the RETRY-INTERVAL setting) until they are returned to sender (based on the RETURN-INTERVAL setting).

SET FORWARD-LOCAL-MAIL

SET FORWARD-LOCAL-MAIL — When TRUE, VSI TCP/IP forwards mail addressed to users on the local host to a central mail hub specified by the *`FORWARDER`* parameter.

Format

`SET FORWARD-LOCAL-MAIL` { TRUE | FALSE }

Description

To configure VSI TCP/IP to direct mail to a central mail hub, you must specify the IP address of the mail hub with the FORWARDER parameter, and define the scope of addresses that you want the mail hub to handle.

By default, when users on the same VSI TCP/IP host send mail to each other, VSI TCP/IP does not route the messages through the mail hub. When FORWARD-LOCAL-MAIL is TRUE, VSI TCP/IP forwards local mail to the mail hub.

To exclude a specific user from the local mail-forwarding system, add the following type of mail alias to `IP$SMTP_ALIASES`:

```
username : *;
```

SET FORWARD-REMOTE-MAIL

SET FORWARD-REMOTE-MAIL — When TRUE, VSI TCP/IP forwards mail addressed to non-local users on a central mail hub specified by the *`FORWARDER`* parameter.

Format

`SET FORWARD-REMOTE-MAIL` { TRUE | FALSE }

Description

To configure VSI TCP/IP to direct mail to a central mail hub, you must specify the IP address of the mail hub with the FORWARDER parameter, and define the scope of addresses that you want the mail hub to handle.

By default, when VSI TCP/IP users send mail to users on other hosts, VSI TCP/IP does not route the messages through the mail hub. When FORWARD-REMOTE-MAIL is TRUE, VSI TCP/IP forwards non-local mail to the mail hub.

By default VSI TCP/IP considers all remote hosts non-local. You can add hosts in other domains to the local-domain list with the **ADD LOCAL-DOMAIN** command.

SET HEADER-CONTROL

SET HEADER-CONTROL — Specifies which RFC-822 message headers are included in messages delivered to local OpenVMS MAIL users.

Format

`SET HEADER-CONTROL` *`header_type`*

Parameter

`header_type`

Either NONE, MAJOR, or ALL.

- NONE eliminates the RFC-822 message headers from locally delivered OpenVMS MAIL messages.
- MAJOR (the default) includes all but Received and Return Path headers.
- ALL includes all headers.

SET HOST-ALIAS-FILE

SET HOST-ALIAS-FILE — Specifies a file from which VSI TCP/IP obtains a list of host aliases. A common use for SMTP host names is when your system is a member of a homogeneous

OpenVMScluster, and you want all mail from any cluster member to appear to be from the same host (for example, the cluster alias). Unlike the MAIL-CONFIG **SET SMTP-HOST-NAMES** command which has a limit of 16 host names, **SET HOST-ALIAS-FILE** lets you specify a host alias file containing as many host aliases as needed. The host name or alias you specify should be registered in the Domain Name System or in the host tables of any system to which you send mail; otherwise, the recipients of your mail will be unable to reply to it. If this logical name is not defined, the SMTP software looks for the file `IP$SMTP_HOST_ALIASES` by default.

Format

```
SET HOST-ALIAS-FILE file_spec
```

Parameter

file_spec

Specifies the file that contains a list of SMTP host names.

SET LOCAL-MAIL-FORWARDER

SET LOCAL-MAIL-FORWARDER — Forwards failed local mail to a specific host.

Format

```
SET LOCAL-MAIL-FORWARDER hostname
```

Parameter

hostname

Specifies the name of the host to which failed local mail is directed.

SET POSTMASTER

SET POSTMASTER — Identifies the user responsible for mail on the system.

Format

```
SET POSTMASTER [username]
```

Parameter

[*username*]

Specifies the name of the user who will receive messages addressed to Postmaster on the local host. If omitted, the user name POSTMASTER is used.

To assign multiple users as the postmaster, enter POSTMASTER, then create an alias for postmaster in the alias file. For example, to make both "username1" and "username2" postmasters, enter the following line in the alias file:

```
postmaster   : username1, username2;
```

SET QUEUE-COUNT

SET QUEUE-COUNT — Specifies the number of mail processing queues that should be created on a system.

Format

`SET QUEUE-COUNT` *`node_name`* [*`count`*]

Parameter

`node_name`

Specifies the OpenVMScluster (SCS) node name of the node whose queue count you want to set, or specifies DEFAULT to set the default for all nodes not specifically set. In a non-cluster environment, only the DEFAULT setting is used.

[*`count`*]

Specifies the number of queues to create on the specified node. If a count is omitted, the queue-count setting for the specified node is removed.

SET REPLY-CONTROL

SET REPLY-CONTROL — Specifies how Internet mail headers are mapped to the OpenVMS MAIL "From" header.

Format

`SET REPLY-CONTROL` [*`hdr_types`*]

Parameter

[*`hdr_types`*]

Specifies a comma-delimited list of SMTP headers (ENVELOPE-FROM, FROM, or REPLY-TO) that are mapped to the OpenVMS MAIL "From" header. The default is "ENVELOPE-FROM, FROM, REPLY-TO."

SET RESENT-HEADERS

SET RESENT-HEADERS — When FALSE, the VSI TCP/IP SMTP symbiont omits the Resent-From, Resent-To, and Resent-Date headers that are usually included when a message is forwarded using a OpenVMS MAIL forwarding address. The default is TRUE. Use this option if mail user agents at your site cannot properly distinguish between normal "From" headers and "Resent-From" headers.

Format

`SET RESENT-HEADERS` { TRUE | FALSE }

SET RETRY-INTERVAL

SET RETRY-INTERVAL — Specifies the amount of time that elapses before another attempt is made to send a message after a failed attempt.

Format

`SET RETRY-INTERVAL` [*`interval`*]

Parameter

[*`interval`*]

Specifies the interval, in minutes (by default, 30 minutes).

SET RETURN-INTERVAL

SET RETURN-INTERVAL — Specifies the amount of time that a message can remain in the processing queue before it is returned to the sender.

Format

[SET RETURN-INTERVAL] [*`interval`*]

Parameter

[*`interval`*]

Specifies the interval, in hours; by default, 96 (four days). A message typically only remains in the processing queue if it cannot be sent over the network to a remote host. When such a message is returned to its sender, the returned message includes the reason why it could not be sent.

SET SEND-BROADCAST-CLASS

SET SEND-BROADCAST-CLASS — Specifies the broadcast class to use to deliver immediate (SEND) messages.

Format

`SET SEND-BROADCAST-CLASS` [*`class_number`*]

Parameter

[*`class_number`*]

Specifies the class-number in a range from 1 to 16, corresponding to the OpenVMS USER1 through USER16 broadcast classes (by default, 16).

SET SMTP-HOST-NAMES

SET SMTP-HOST-NAMES — Sets the host name from which all outgoing mail appears to be sent and the aliases for which this host accepts incoming mail. A common use for SMTP HOST NAME is when your system is a member of a homogeneous OpenVMScluster, and you want all mail from any cluster member to appear to be from the same host.

Format

SET SMTP-HOST-NAMES *host_names*

Parameter

host_names

Contains a comma-delimited list of host names. The first name in the list specifies the host name from which all outgoing mail appears to be sent. The remaining host names in the list specify the aliases for which this host accepts incoming mail.

Note

The specified host name or alias should be registered in the Domain Name System or in the host tables of any system that you send mail to; otherwise, the recipients of your mail will be unable to reply to it.

SET START-QUEUE-MANAGER

SET START-QUEUE-MANAGER — Determines whether `START_SMTP.COM` starts the OpenVMS queue manager if it is not already running. The default is TRUE.

Format

SET START-QUEUE-MANAGER { TRUE | FALSE }

SHOW

SHOW — Displays the current configuration.

Format

SHOW

SPAWN

SPAWN — Executes a single DCL command, or if entered without options, starts a subprocess with the same effect as PUSH. To return from DCL, use the **LOGOUT** command. If the `IP $DISABLE_SPAWN` logical is set, **SPAWN** does not work.

Format

`SPAWN` [*`command`*]

Parameter

[*`command`*]

Specifies a command to execute. If you omit command, a DCL command line subprocess is created.

Qualifiers

`/INPUT=`*`file-spec`*

Specifies an input file to the command you enter with **SPAWN**.

`/LOGICAL_NAMES`

`/NOLOGICAL_NAMES`

Specifies that logical names and logical name tables are not copied to the subprocess.

`/SYMBOLS`

`/NOSYMBOLS`

Specifies that global and local names are not passed to the subprocess.

`/WAIT`

`/NOWAIT`

Returns control without waiting for the command to complete. Do not use this qualifier with commands that have prompts or screen displays.

`/OUTPUT=`*`file-spec`*

Specifies a file that retains the output of the command invoked with **SPAWN**. This qualifier only works when a single command is entered without creating a DCL subprocess. In addition, this qualifier is positional; you must enter it immediately after **SPAWN** or other qualifiers.

STATUS

STATUS — Indicates whether the SMTP configuration has been modified.

Format

`STATUS`

USE

USE — Reads in a VSI TCP/IP SMTP configuration file. After a **USE**, you can use the various configuration commands to modify the SMTP configuration. (Functionally equivalent to **GET**.)

Format

```
USE config_file
```

Parameter

`config_file`

Specifies the name of the SMTP configuration file to read in.

VERSION

VERSION — Displays the MAIL-CONFIG version and release information.

Format

```
VERSION
```

WRITE

WRITE — Writes the current VSI TCP/IP SMTP configuration to SMTP configuration files. (Functionally equivalent to **SAVE**.)

Format

```
WRITE config_file
```

Parameter

`config_file`

Specifies the name of the file to which to write the current VSI TCP/IP SMTP configuration. By default, the configuration is saved to the same file from which it was read.

Chapter 4. NET-CONFIG Command Reference

This chapter describes the commands you can run from the NET-CONFIG command line. NET-CONFIG lets you examine, modify, and save configuration files for VSI TCP/IP network interfaces.

To invoke NET-CONFIG:

```
$ IP CONFIGURE /INTERFACES
```

At any NET-CONFIG prompt, you can list the available commands by typing "?". Online help for each NET-CONFIG command is available through the NET-CONFIG **HELP** command.

Of the **SET** commands, the **SET WINS-COMPATIBILITY** command requires that you reboot the system after you use the command; all other **SET** commands can be executed without rebooting the system.

For details on configuring network interfaces, refer to the *VSI TCP/IP Administrator's Guide: Volume II*.

Command Summary

Table 4.1 lists the commands you can run from the NET-CONFIG prompt.

Table 4.1. NET-CONFIG Command Summary

NET-CONFIG Command	Description
ADD	Adds a device to the network configuration.
ATTACH	Attaches a terminal to a process.
CHECK	Performs a check on the current configuration.
CLEAR	Deletes all devices from the network configuration.
CREATE	Create a Six-To-Four interface to carry IPv6 traffic over IPv4.
DELETE	Deletes a single device from the network configuration.
DISABLE	Disables a device and deletes it from the configuration.
ENABLE	Enables a device.
ERASE	Deletes all devices from the network configuration.
EXIT	Exits NET-CONFIG and saves the configuration if it changed.
GET	Reads in a nonstandard configuration file.
HELP	Displays command help information.
MODIFY	Changes a device configuration.
PUSH	Accesses the DCL command interpreter.
QUIT	Exits NET-CONFIG and prompts to save the configuration if it changed.
SAVE	Writes configuration parameters to a file.
SET ANONYMOUS-FTP-ACCESS	Restricts the ANONYMOUS FTP user permissions to list, read, write, or delete files.

NET-CONFIG Command	Description
SET ANONYMOUS-FTP-DIRECTORY	Restricts directory tree access for the ANONYMOUS FTP user.
SET CLUSTER-SERVICE-ADDRESS	Identifies the IP address to use when responding to address queries from hosts specified by *`CLUSTER-SERVICE-NAMES`*.
SET CLUSTER-SERVICE-NAMES	Lists the host names to use for load balancing.
SET DEFAULT-RMT-TAPE-DEVICE	Defines the default RMT server tape device.
SET DEFAULT-ROUTE	Defines the default IP route.
SET DOMAIN-NAMESERVERS	Configures the DNS domain resolver.
SET HOST-NAME	Defines the local host name and domain.
SET IP-CLUSTER-ALIASES	Specifies a list of cluster-wide Internet addresses to which this node responds.
SET LOAD-PWIP-DRIVER	Enables automatic loading of the version 5 PATHWORKS server interface (PWIP) driver when VSI TCP/IP starts
SET LOCAL-DOMAIN	Overrides the default local domain derived from previous use of the **SET HOST-NAME** command.
SET LPD-DEFAULT-USERNAME	Specifies a default user name for print jobs received from a remote machine via the LPD protocol.
SET NAMESERVER-RETRANSMISSION	Specifies a time between and the number of name server requests that are made before the system stops sending name server requests to a nonresponding server.
SET SPOOL-DIRECTORY	Sets the default spool directory name.
SET TFTP-DIRECTORY	Sets the default directory for the TFTP server.
SET TIMEZONE	Sets the local timezone.
SET TIMEZONE-RULES	Specifies the timezone rules file name.
SET WHOIS-DEFAULT-SERVER	Sets the default WHOIS server.
SET WINS-COMPATIBILITY	Defines the logical names used with the WIN/TCP and Pathway products from The Wollongong Group.
SHOW	Displays the current configuration or device names.
SPAWN	Executes a DCL command.
STATUS	Displays the device count in the configuration.
USE	Reads in a nonstandard configuration file.
VERSION	Displays the NET-CONFIG and OpenVMS versions.
WRITE	Writes the current configuration.

ADD

ADD — Adds a device to the network configuration.

Format

```
ADD interface
```

Parameters

interface

Specifies the name of the device interface to add.

Refer to Table 4.2 for a list of supported network interfaces and the corresponding interface parameters for which you are prompted. For descriptions of interface parameters, refer to Table 4.3.

Table 4.2. Interfaces and Parameters

Type	**Description**	
nsip	**Interface name**: nsip0, nsip1, ... nsip9 **Device type**: IPX-over-IP tunnel	
	Parameter Prompt	**Example Value**
	IP address of remote system:	192.41.228.70
	The nsip interface provides access to IPX-over-IP for connecting to Novell's IP tunnel feature.	
pd	**Interface name:** pd0 **Device type:** Secondary Ethernet Address	
	Parameter Prompt	**Example Value**
	TCP/IP	161.44.128.21
	Hardware-Device	se0
	IP SubNet Mask	None
	Non-Standard IP Broadcast Address:	None
	pd0 (Secondary Ethernet Address):	Csr=None, Flags=%X0
	IPv6 on this interface [DISBALED]:	ENABLE
	IPv6 global address [NONE]:	xxxx:xxxx::xxxx:xxxx
	IPv6 mask length:	16
	Careless assignment of a secondary address can cause network problems. In general, you should assign pseudo devices (pd) addresses on the same network or subnet as the se device to which the pd device is linked. If the pd interface is not in the same IP network as its associated se interface, some TCP/IP packages (such as early versions of SunOS) retransmit broadcast packets for the other IP network back to the network segment from which they were transmitted. This can cause network storms.	

Type	Description	
	Note GateD will listen to traffic on pseudo devices in VSI TCP/IP v10.5. Some services listen to traffic on se interfaces only and ignore traffic on pd interfaces. One such service is the RIP listener in GATED.	
ppp	**Interface name**: ppp0, ... **Device type**: Any supported PPP terminal interface	
	Parameter Prompt	**Example Value**
	VMS Device:	TTA0:
	Baud Rate:	19200
	PPP Authentication Method:	None
	PPP Protocol Compression:	OFF
	PPP Address and Control Field Compression:	OFF
	PPP Retry Count:	0
	PPP Idle Timeout:	0
	PPP MRU Size:	0
	PPP ICMP:	ENABLED
	PPP TCP Compression:	OFF
	PPP Termination Retry Count:	0
	PPP Timeout:	0
	IP Address:	0.0.0.0
	Point-to-Point Device IP Destination Address:	0.0.0.0
	IP Subnet Mask:	255.0.0.0
rp	**Interface name**: rp0, rp1, rp2, ... **Device type**: Raw packet	
	Parameter Prompt	**Example Value**
	IP Address:	192.41.228.70
	IP SubNet Mask:	255.255.255.0
	The rp interface allows IP packets that are normally destined for transmission to be directed instead to a user process by way of an AF_RAWPACKET socket.	
se	**Interface name**: se0, se1, se2, ... **Device type**: Any HP VMS Ethernet, FDDI, or Token-Ring Alpha controller	
	Parameter Prompt	**Example Value**
	VMS Device:	XEA0
	Link Level Encapsulation Mode:	ETHERNET
	BSD Trailer Encapsulation:	DISABLED

Type	Description	
	IP Address:	192.41.228.70
	IP SubNet Mask:	255.255.255.0
	Non-Standard IP Broadcast Address:	192.41.228.71
	DHCP CLIENT [DISABLED]:	DISABLED
	Jumbo Frames [DISABLED]:	ENABLED
	IPv6 on this interface [DISABLED]:	ENABLED
	IPv6 global address [NONE]:	3FFE:1200:3006::C673:8EBE
	IPv6 mask length:	48
	These interface uses any HP Ethernet controller to provide access to a 10/100/1000 Mb/s Ethernet network, any HP FDDI controller to provide access to a 100 Mb/s FDDI network, and an HP Token-Ring Alpha controller to provide access to a 4 Mb/s or 16 Mb/s Alpha Token-Ring. The se interface uses the standard OpenVMS Ethernet driver to allow VSI TCP/IP to share the Ethernet devices with other protocols such as LAT, LAVC, and DECnet.	
sl	**Interface name**: sl0, sl1, sl2, ... **Device type**: Any OpenVMS-supported terminal interface	
	Parameter Prompt	**Example Value**
	VMS Device:	TTA0
	Baud Rate:	19200
	Header Compression Mode:	DISABLED
	IP Address:	192.41.228.70
	Point-To-Point Device IP Destination Address:	192.41.228.71
	IP SubNet Mask:	255.255.255.0
	The VSI TCP/IP software supports SLIP with Van Jacobson's header compression algorithm, reducing the size of the headers and increases the bandwidth available to data. Header compression mode is determined by what both sides can support.	

Table 4.3. NET-CONFIG Prompts

NET-CONFIG Prompts	Function
ACCM Mask	Asynchronous Control Character Map Mask. A 32-bit mask that indicates the set of ASCII control characters to be mapped into two-character sequences for transparent transmission over the line. The default is %x00000000.
Address and Control Field Compression (ACFC)	When ON, PPP eliminates the address and control fields when they are identical over a series of frames. The default is OFF.
Authentication Method	Determines the type of remote peer authentication required to allow network- layer protocol packets to be exchanged. Accepted values are "PAP" (Password Authentication Protocol) and "NONE" (the default).

NET-CONFIG Prompts	Function
Baud Rate	Indicates the baud rate of transmission. Possible values are: 110, 300, 1200, 2400, 4800, 9600, 19200, and UNSPECIFIED.
BSD Trailer Encapsulation	For 10Mb/sec Ethernet controllers only. Can be used to enable Berkeley Trailer encapsulation of IP packets on those Ethernets. There are two possible settings: NEGOTIATED or DISABLED. The default, DISABLED, prevents the use of trailer encapsulation.
Hardware Device	The name of the real Ethernet device; for example, se0.
Header Compression Mode	For PPP and SLIP devices, indicates whether to use Van Jacobson's TCP header compression algorithm. The parameter has three possible settings: DISABLED, ENABLED, or NEGOTIATED. DISABLED indicates that headers should never be compressed. ENABLED indicates that headers should always be compressed. The default is DISABLED. NEGOTIATED indicates that headers should not be compressed until a compressed header is received from the other side. At least one side of a link must be ENABLED for compression to be used; that is, both sides of a link cannot be set to NEGOTIATED for compression to be used.
ICMP	When ENABLED (the default), PPP allows ICMP packets over the PPP connection. Administrators may want to disable ICMP packets if they are concerned with "service attacks" from dial-up connections.
Idle Timeout	Determines how long (in seconds) the connection must be idle before PPP attempts to close it with "Terminate-Request" packets. The default is 0.
IP Address	Indicates the Internet address, in dotted decimal notation, associated with the interface. For PPP interfaces, you can specify 0.0.0.0 to indicate that the local IP address will be specified by the remote peer when a serial connection is established. The default is 0.0.0.0.
IP Address of remote system	Indicates the Internet address associated with the remote system.
IP Broadcast Address	For devices that support broadcasts, allows the setting of a non-standard IP broadcast address. The parameter defaults to an address whose host portion is all ones.
IP Subnet Mask	The subnet mask of the local interface in dotted decimal format. The default depends on the local interface IP address. For example, a class A address results in a default subnet mask of 255.0.0.0.
IPv6 global address	Indicates the global unique address associated with this interface. The interface may also have a link-local address which will be automatically generated when the interface is started
IPv6 mask length	The length of the mask for the IPv6 address.
Link Level Encapsulation Mode	ETHERNET
Maximum Receive Unit (MRU) Size	Determines the maximum number of 8-bit bytes for the PPP Information field, including Padding, but not including the Protocol field. Because opposite ends of a PPP connection may have different MRU values, PPP negotiates a suitable MRU for both systems. The default is 500.

NET-CONFIG Prompts	Function
Point-To-Point Device IP Destination Address	For point-to-point interfaces, indicates the IP address of the peer system on the other side of the connection. The default is 0.0.0.0.
Protocol Compression	When ON, PPP negotiates with the peer to use one byte instead of two for the Protocol fields to improve transmission efficiency on low-speed lines. Default is OFF.
Retry Count	Determines the number of attempts PPP makes to configure a connection with "Configure-Request" packets. The default is 0.
Termination Retry Count	Determines the number of attempts PPP makes to terminate a connection with "Terminate-Request" packets. The default is 0.
Timeout	Determines the time (in seconds) between successive Configure-Request or Terminate-Request packets. The default is 0.
VMS Device	For devices that use a OpenVMS device driver to interface with the hardware, indicates the name of the OpenVMS device that VSI TCP/IP will use. This parameter is used with HP Ethernet, PPP, and SLIP interfaces.

Example

```
$ IP CONFIGURE /NETWORK
VSI TCP/IP NFS Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG> ADD PPP2
[Adding new configuration entry for device "PPP2"]
VMS Device: [TTA0]
Baud Rate: [UNSPECIFIED] 9600
PPP ACCM Mask: [%x0]
PPP Authentication Method: [NONE] PAP
PPP Protocol Compression: [OFF] ? ON
PPP Address and Control Field Compression: [OFF] ON
PPP Retry Count: [0] 10
PPP Idle Timeout: [0] 10
PPP MRU Size: [0]
PPP ICMP: [ENABLED]
PPP TCP Compression: [OFF]
PPP Termination Retry Count: [0] 10
PPP Timeout: [0] 10
IP Address: [NONE] 123.45.67.00
Point-to-Point Device IP Destination Address: [NONE]
IP Subnet Mask: [NONE]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
```

ATTACH

ATTACH — Detaches the terminal from the calling process and reattaches it to another process. Use the **SPAWN SHOW PROCESS /SUBPROCESSES** command to list the names of subprocesses. Use the DCL **LOGOUT** command to return to the original process. If the `IP$DISABLE_SPAWN` logical is enabled, **ATTACH** does not work.

Format

`ATTACH` *`process-name`*

Parameter

`process_name`

Specifies the name of a process to which you want our terminal attached. (Not all subprocesses can be attached; some testing may be required.)

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SPAWN
$ MM
MM>SPAWN SHOW PROCESS /SUB
...
There are 3 processes in this job:
_TWA42:
 PROC_1
   PROC_2 (*)
MM>ATTACH _TWA42:
NET-CONFIG>ATTACH PROC_1
MM>QUIT
$ LOGOUT
NET-CONFIG>
```

This example shows the use and exit of attached subprocesses.

1. The first command uses **SPAWN** to create a subprocess. **MM** is invoked from the DCL com-mand line. Next, the **SPAWN SHOW PROCESS /SUB** command is used to list all the subprocess names. The display shows that three subprocesses are active. (Process _TWA42: is NET-CONFIG, PROC_1 is **MM**, and **PROC_2** is the **SPAWN SHOW PROCESS/SUB** command.)

2. In the next command, the MM **ATTACH** command returns control to the NET-CONFIG process. From this utility, **ATTACH** returns control to MM. To exit, **QUIT** is invoked from **MM**, and **LOGOUT** is invoked at the original spawned DCL command line; control returns back to NET-CONFIG. (If **SPAWN SHOW PROCESS /SUB** had been entered, only this command and the configuration processes would be active.)

CHECK

CHECK — Checks the configuration parameters to ensure all required information is provided. If a problem is found, an error message displays; otherwise, if the check is successful, no information displays.

Format

`CHECK`

Description

CHECK provides a check of the current configuration. The following messages can display:

Message	Description
ERROR: `device_name` cannot $ASSIGN to FFI device: *value*	The specified X device does not exist, or a lack of privileges made opening the device impossible.
ERROR: `device_name` cannot $ASSIGN to SLIP device: `value`	The specified device does not exist, or a lack of privileges made opening the device impossible.
ERROR: `device_name`: Default route cannot be the local machine: `default_route`	The default route is the gateway that connects this system to the Internet; it cannot be the local system.
ERROR: `device_name`: Default route route must be directly connected	The IP address specified in the **SET DEFAULT-ROUTE** command must be for a system connected to the Internet.
ERROR: `device_name`: Illegal value for IP Address: `address`	The specified IP address either has an octet above 255, or contains an illegal character such as a space, an alphabetical character, or a control character. Use the **MODIFY** command to specify a new value.
ERROR: `device_name`: Illegal value for IP Broadcast Address: `address`	The specified IP broadcast address either has an octet above 255, or contains an illegal character such as a space, an alphabetical character, or a control character. Use the **MODIFY** command to specify a new value.
ERROR: `device_name`: Illegal value for IP SubNet Mask: `mask`	The specified IP subnet mask value either has an octet above 255, or contains an illegal character such as a space, an alphabetical character, or a control character. Use the **MODIFY** command to specify a new value.
ERROR: `device_name` is slave to nonexistent device `hardware_device`	The VSI TCP/IP device points to a device that does not exist for the local system.
ERROR: `device_name`'s DECnet peer must be DECnet node name, not address	DECnet node names are alphanumeric strings of six characters or less; specify the correct value.
ERROR: `device_name`'s point-to-point destination is unspecified	A destination address was not specified for a point-to-point device, such as a SLIP line.
ERROR: `device_name`'s PSI peer must be PSI DTE, not node.	The "IP Over PSI Peer Host's DTE" prompt is requesting the DTE destination name; specify the correct value.
ERROR: `device_name's` PSI local must be PSI DTE, not node.	The "IP Over PSI Local Host's DTE" prompt is requesting the DTE source name; specify the correct value.
ERROR: The host name is not a domain-style host name and domain name service is enabled.	The domain name did not contain dot separators.
ERROR: There is no Host Name specified.	A host name value was not entered either in the configuration file you are creating or in a read-in configuration file.

Message	Description
WARNING: *`device_name`* has no FFI device specified	The added or modified device requires that an FFI device be specified.
WARNING: *`device_name`* has no protocol addresses specified	The added or modified device requires that protocol addresses be specified.
WARNING: *`device_name`*'s DECnet peer is unspecified (link will not come up)	Both ends of a point-to-point DECnet link must be specified.
WARNING: *`device_name`*'s hardware device is unspecified	A device name must be entered for this device.
WARNING: *`device_name`*'s PSI peer is unspecified (link will not come up)	Both ends of a point-to-point PSI link must be specified.
WARNING: *`device_name`*'s PSI local DTE is unspecified (link will not come up)	Both ends of a point-to-point PSI link must be specified.

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>ADD PPP1
[Adding new configuration entry for device "ppp1"]
VMS Device: [TTA0]  TTA1
Baud Rate:  [UNSPECIFIED] 9600
PPP ACCM Mask:  [%x0]
PPP Authentication Method: [NONE] PAP
PPP Protocol Compression: [OFF] ON
PPP Address and Control Field Compression: [OFF] ON
PPP Retry Count: [0] 10
PPP Idle Timeout: [0] 10
PPP MRU Size: [0] 10
PPP ICMP:  [ENABLED]
PPP TCP Compression: [OFF]
PPP Termination Retry Count: [0]
PPP Timeout: [0]
IP Address: [NONE]
Point-to-Point Device IP Destination Address: [NONE] 155.45.24.13
IP Subnet Mask: [NONE]
[ppp] (Point-to-Point Protocol): Csr=NONE, Flags=%X0]
NET-CONFIG>CHECK
WARNING: ppp1 has no protocol addresses specified
NET-CONFIG>EXIT
$
```

CLEAR

CLEAR — Deletes all devices from the current network configuration.

Format

```
CLEAR
```

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>CLEAR
```

CREATE

CREATE — Create an Six-to-Four IPv6 interface.

Format

```
CREATE
```

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>CREATE SIX-TO-FOUR
IPv4 address to use [none]: 192.168.1.1
Mask Length [48]:
```

Note

RFC 1597 private address (10.*.*.*, 172.16.*.*, 192.168.*.*) and RFC 3927 IPv4 link-local addresses (169.254.*.*) are not allowed for the IPv4 address.

DELETE

DELETE — Removes a single device from the network configuration.

Format

DELETE *interface*

Parameter

interface

Specifies the name of the interface for the device being removed.

Example

This example deletes a SLIP device, SL0, from the current configuration.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>DELETE SL0
```

DISABLE

DISABLE — Disables a device. A disabled device is deleted from the configuration and is not configured at network boot.

Format

`DISABLE` *`interface`*

Parameter

`interface`

Specifies the name of the interface for the device being disabled.

Example

This example disables a SLIP device, SL0.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>DISABLE SL0
```

ENABLE

ENABLE — Enables a device.

Format

`ENABLE` *`interface`*

Parameter

`interface`

Specifies the name of the interface for the device being enabled.

Example

This example enables a SLIP device, SL0.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
```

```
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>ENABLE SL0
```

ERASE

ERASE — Erases all configured devices from the current configuration. (Functionally equivalent to **CLEAR**.)

Format

```
ERASE
```

EXIT

EXIT — Saves the current configuration, if it has been modified, and exits the configuration program. (Use the **STATUS** command to display whether the configuration was modified.)

Format

```
EXIT
```

Examples

When the configuration has not been changed, a message displays indicating that the configuration file is not updated.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>EXIT
[Configuration not modified, so no update needed]
$
```

When the configuration has changed, a message displays indicating that the configuration file has been updated.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>CLEAR
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$
```

GET

GET — Reads in a nonstandard configuration file.

Format

GET *filename*

Parameter

filename

Specifies the name of the configuration file to read; by default, `NETWORK_DEVICES.CONFIGURATION` in the current working directory.

Example

This example retrieves the configuration file `IP$:TEST.CONFIGURATION` into the NET-CONFIG workspace.

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>GET IP$:TEST.CONFIGURATION
```

HELP

HELP — Displays help information by listing either the command names or information about specific commands. You can also display help information by adding a question mark to other NET-CONFIG commands.

Format

HELP *command*

Parameter

command

Specifies the command for which to list help information. Use a question mark (?) to list all command names. Entering a question mark on the NET-CONFIG command line has the same effect as using the **HELP ?** command.

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>HELP ?
ADD    ATTACH   CHECK    CLEAR    DELETE   DISABLE  ENABLE   ERASE
EXIT   GET      HELP     MODIFY   PUSH     QUIT     SAVE     SET
SHOW   SPAWN    STATUS   USE      ERSION   WRITE
NET-CONFIG>
```

MODIFY

MODIFY — Modifies an existing device configuration. If the device has not already been configured, an error message displays. Use **MODIFY** to modify configuration parameters for an existing network interface. **MODIFY** produces the same sequence of prompts for network interface parameter values as the **ADD** command, but uses the current settings for default values.

Format

`MODIFY` *`interface`*

Parameter

`interface`

Specifies the name of the interface for the device being modified.

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>MODIFY SE0
[Modifying configuration entry for device "se0"]
VMS Device: [EZA0]
Link Level Encapsulation Mode: [ETHERNET]
BSD Trailer Encapsulation: [DISABLED]
IP Address: [123.45.678.90]
IP Subnet Mask: [255.255.255.0]
Non-Standard IP Broadcast Address: [NONE]
se0 (Shared OpenVMS Ethernet/FDDI): Csr=NONE, flags=%x0)
NET-CONFIG>
```

PUSH

PUSH — Starts and attaches a DCL subprocess. If a parent process exists, attach to it. To return from DCL, use the **ATTACH** or the **LOGOUT** command. To switch back from a DCL subprocess, use the **ATTACH** command. If the `IP$DISABLE_SPAWN` logical is set, **PUSH** does not work.

Format

```
PUSH
```

QUIT

QUIT — Prompts you to save the current configuration if it was modified, and then exits.

Format

```
QUIT
```

Example

```
$ IP CONFIGURE /NETWORK_DEVICES
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>DISABLE SL0
NET-CONFIG>QUIT
Configuration modified, do you want to save it?  [NO]
```

SAVE

SAVE — Saves the configuration parameters.

Format

`SAVE /STARTUP` *`filename`*

Parameter

`filename`

Specifies the file name for the configuration file or the startup command procedure. The default for filename when saving the configuration data is the file from which the configuration was read. The default for filename when saving the VSI TCP/IP startup command procedure is `IP$SYSTARTUP.COM`.

Qualifier

`/STARTUP`

Specifies that NET-CONFIG saves the VSI TCP/IP startup commands in filename.

SET ANONYMOUS-FTP-ACCESS

SET ANONYMOUS-FTP-ACCESS — Defines file access rights for the ANONYMOUS FTP user to read, write, delete, and list files in addition to spawning a subprocess. If your configuration permits users to make anonymous file transfers via the OpenVMS ANONYMOUS account, use the *`ANONYMOUS-FTP-ACCESS`* parameter to restrict file access for that account. You can change the value of *`ANONYMOUS-FTP-ACCESS`* without rebooting by also defining or redefining the system-wide `IP$ANONYMOUS_FTP_CONTROL` logical name.

Format

`SET ANONYMOUS-FTP-ACCESS` [*`access`*]

Parameters

[*`access`*]

NOLIST

Disables the listing of files.

NOWRITE

Disables the storing of files.

NOSPAWN

Disables the SPAWN command.

NOREAD

Disables reading of files.

NODELETE

Disables the deleting/renaming of files.

By default, the NOWRITE, NOSPAWN settings are used for anonymous FTP sessions.

Note

To cancel a restriction, omit the appropriate option from the parameter list. To grant all file access rights, run the **SET ANONYMOUS-FTP-ACCESS** command without parameters.

Example

The following example sets the *ANONYMOUS-FTP-ACCESS* parameter to NOWRITE, and then sets the system-wide IP$ANONYMOUS_FTP_CONTROL logical name. By setting this logical name, the *ANONYMOUS_FTP_ACCESS* parameter can be set without rebooting the system. Setting to NOWRITE prevents anonymous users from storing files.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP Network Configuration Utility V10.5
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET ANONYMOUS-FTP-ACCESS NOWRITE
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next VSI TCP/IP reload]
$ DEFINE /SYSTEM /EXECUTIVE_MODE -
_$ IP$ANONYMOUS_FTP_CONTROL "NOWRITE"
```

SET ANONYMOUS-FTP-DIRECTORY

SET ANONYMOUS-FTP-DIRECTORY — Restricts directory tree access for the ANONYMOUS FTP user. If your configuration permits users to make anonymous file transfers via the OpenVMS ANONYMOUS account, use the *ANONYMOUS-FTP-DIRECTORY* parameter to restrict access for that account to a specific directory tree. You can change the value of *ANONYMOUS-FTP-DIRECTORY* without rebooting by also defining or redefining the system-wide

`IP$ANONYMOUS_FTP_DIRECTORY` logical name.

Format

`SET ANONYMOUS-FTP-DIRECTORY` [*directory*]

Parameter

[*directory*]

Specifies the name of the directory tree to which the account is restricted. To cancel the restriction, enter the command without a directory name.

Example

This example sets the *ANONYMOUS-FTP-DIRECTORY* parameter to `USERS:[ANONYMOUS]`, then sets the system-wide `IP$ANONYMOUS_FTP_DIRECTORY` logical name. By setting the system-wide logical name, *ANONYMOUS-FTP-DIRECTORY* can be set without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET ANONYMOUS-FTP-DIRECTORY USERS:[ANONYMOUS]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE -
_$ IP_ANONYMOUS_FTP_DIRECTORY "USERS:[ANONYMOUS]"
$
```

SET CLUSTER-SERVICE-ADDRESS

SET CLUSTER-SERVICE-ADDRESS — Identifies the IP address to be used when responding to address queries for hosts specified by *CLUSTER-SERVICE-NAMES*. If your host is configured with more than one interface, the DOMAINNAME service is enabled, and you are using cluster load balancing, use the *CLUSTER-SERVICE-ADDRESS* parameter to identify the IP address that this host will return in response to address queries for any host specified by *CLUSTER-SERVICE-NAMES*. If you do not specify an address, an interface is chosen at random.

Format

`SET CLUSTER-SERVICE-ADDRESS` [*IP_address*]

Parameter

[*IP_address*]

Specifies the IP address returned when responding to address queries from hosts identified by CLUSTER-SERVICE-NAMES.

Example

This example sets the *`CLUSTER-SERVICE-ADDRESS`* parameter to 191.43.154.10 and then sets the system-wide `IP$CLUSTER_SERVICE_ADDRESS` logical name. By setting the system-wide logical name, the *`CLUSTER-SERVICE-ADDRESS`* can be set without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET CLUSTER-SERVICE-ADDRESS 191.43.154.10
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE -
_$ IP_CLUSTER_SERVICE_ADDRESS "191.43.154.10"
$ @IP$:START_SERVER
$
```

SET CLUSTER-SERVICE-NAMES

SET CLUSTER-SERVICE-NAMES — Lists host names to which cluster load balancing applies, and allows you to configure hosts so that TCP-based connections are directed to the host with the lightest load at the time of the request.

Format

`SET CLUSTER-SERVICE-NAMES` *`hostlist`*

Description

The DOMAINNAME service must be enabled on each host. To establish this environment:

1. Configure the primary name server for the parent domain so that it delegates authority for each host specified with the *`CLUSTER-SERVICE-NAMES`* parameter to this host.

2. Use the *`CLUSTER-SERVICE-NAMES`* parameter to identify the hosts for which this host will accept connections.

For example, if authority for `CLUSTER.FLOWERS.COM` is delegated to NODE1, NODE2, and NODE3 on the primary name servers, each node must include `CLUSTER.FLOWERS.COM` in its *`CLUSTER-SERVICE-NAMES`* list. When a TCP service request is made to `CLUSTER.FLOWERS.COM`, DNS returns the IP addresses of the nodes NODE1, NODE2, or NODE3, ordered by load rating.

If the host is configured with more than one interface, specify the desired interface with the *`CLUSTER-SERVICE-ADDRESS`* parameter.

Parameter

`hostlist`

Specifies a comma-separated list of fully qualified host names for which cluster load balancing is implemented.

Example

This example sets the *`CLUSTER-SERVICE-NAMES`* parameter to `CLUSTER.FLOWERS.COM`, then sets the system-wide `IP$CLUSTER_SERVICE_NAMES` logical. By setting the system-wide logical, *`CLUSTER-SERVICE-NAMES`* can be set without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET CLUSTER-SERVICE-NAMES CLUSTER.FLOWERS.COM
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE/SYSTEM/EXECUTIVE
_$ IP_CLUSTER_SERVICE_NAMES "CLUSTER.FLOWERS.COM"
$ @IP$:START_SERVER
$
```

SET DEFAULT-RMT-TAPE-DEVICE

SET DEFAULT-RMT-TAPE-DEVICE — Defines the default OpenVMS tape drive that the RMT server uses. (This drive is the OpenVMS equivalent of the UNIX /dev/rmt0 interface, and is the drive used by the rdump and rrestore programs.) If the parameter is not set, the server searches for interface types in the following order and uses the first interface it finds: MU, MK, MF, MT, and MS. You can change the value of *`DEFAULT-RMT-TAPE-DEVICE`* without rebooting by also defining or redefining the system-wide logical name `IP$RMT_TAPE_DEVICE`.

Format

`SET DEFAULT-RMT-TAPE-DEVICE` *`drive_name`*

Parameter

`drive_name`

Specifies the name of the default drive.

Example

This example sets *`DEFAULT-RMT-TAPE-DEVICE`* to HSC001$MUA1:, then sets the `IP$RMT_TAPE_DEVICE` system-wide logical name. By setting this logical name, the tape device can be given a new value without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
```

```
NET-CONFIG>SET DEFAULT-RMT-TAPE-DEVICE HSC001$MUA1:
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$RMT_TAPE_DEVICE "HSC001$MUA1:"
$
```

SET DEFAULT-ROUTE

SET DEFAULT-ROUTE — Defines the default IP route-that is, the IP address of the gateway that VSI TCP/IP uses for all packets sent from the local network.

Format

SET DEFAULT-ROUTE *IP_address*

Description

You can specify more complex routing information by either:

- Creating a `IP$:LOCAL_ROUTES.COM` file that contains a list of **IP SET /ROUTE** commands that set up the routing tables on a per-network basis.
- Running a dynamic-routing protocol such as RIP, HELLO, EGP, or BGP by configuring the GATED service.

Note

Any GATED configuration overrides a default IP route set with NET-CONFIG. Once started, GATED takes complete control of your routing.

You can change the default route without rebooting by using the **IP SET /ROUTE** command to delete the old default route and add the new default route.

Parameter

IP_address

Specifies the IP address of the gateway.

Example

This example sets the gateway DEFAULT-ROUTE to 192.41.228.100, then performs the commands that change DEFAULT-ROUTE without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET DEFAULT-ROUTE 192.41.228.100
```

```
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ IP SET/ROUTE  -
_$ /DELETE=(DESTINATION=0.0.0.0, GATEWAY=192.41.228.100)
$ IP SET/ROUTE -
_$ /ADD=(DESTINATION=0.0.0.0, GATEWAY=192.41.228.100)
```

SET DOMAIN-NAMESERVERS

SET DOMAIN-NAMESERVERS — Configures the DNS domain resolver, the portion of VSI TCP/IP called when a host name must be translated into an Internet address.

Format

`SET DOMAIN-NAMESERVERS` [*`address`* [*`address . . .`*]]

Description

If you are using DNS, use this parameter to define the Internet addresses of the Domain Name Servers to which the resolver will send requests.

The usual setting for the parameter is the loopback address (127.0.0.1), which directs the resolver to send inquiries to the server on the local system.

To disable DNS and use the host tables instead of the service, enter the **SET DOMAIN-NAMESERVERS** command with no options.

You can change the value of *`DOMAIN-NAMESERVERS`* without rebooting by also defining or redefining the system-wide logical name `IP$NAMESERVERS` and restarting the IP$SERVER (`@IP$:START_SERVER`) and `SMTP_SYMBIONT` (`@IP$:START_SMTP`) processes.

Parameter

[*`address`* [*`address . . .`*]]

Specifies the IP address of a name server. When you list multiple addresses, the resolver successively attempts to send a packet to the addresses, in the listed order, until it receives a response.

Restriction

The resolver nameserver list can only include three IP addresses. This list is controlled by the `IP$NAMESERVERS` logical (which is controlled by the *`DOMAIN-NAMESERVERS`* parameter).

Examples

This example tells the resolver to try only the local nameserver. The subsequent DCL commands change domain nameservers without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
```

```
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET DOMAIN-NAMESERVERS 127.0.0.1
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXEC IP$NAMESERVERS "127.0.0.1"
$ DEFINE /SYSTEM /EXEC UCX$BIND_SERVER000 "127.0.0.1"
$ DEFINE /SYSTEM /EXEC UCX$BIND_DOMAIN "FLOWERS.COM"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_SERVER000 "127.0.0.1"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_DOMAIN "FLOWERS.COM"
$ @IP$:START_SERVER
$ @IP$:START_SMTP
$
```

This example configures the resolver to try the local name server and, if it fails, to try 192.0.0.1.

```
NET-CONFIG>SET DOMAIN-NAMESERVERS 127.0.0.1,192.0.0.1
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$
$ DEFINE /SYSTEM /EXEC IP$NAMESERVERS "127.0.0.1", "192.0.0.1"
$ DEFINE /SYSTEM /EXEC UCX$BIND_SERVER000 "127.0.0.1"
$ DEFINE /SYSTEM /EXEC UCX$BIND_SERVER001 "192.0.0.1"
$ DEFINE /SYSTEM /EXEC UCX$BIND_DOMAIN "FLOWERS.COM"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_SERVER000 "127.0.0.1"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_SERVER001 "192.0.0.1"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_DOMAIN "FLOWERS.COM"
$ @IP$:START_SERVER
$ @IP$:START_SMTP
$
```

The **SET DOMAIN-NAMESERVERS** disables the local DNS domain server; only host tables are used to translate names and addresses.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET DOMAIN-NAMESERVERS
NET-CONFIG>EXIT
$
$ DEASSIGN /SYSTEM /EXEC IP$NAMESERVER
$ DEASSIGN /SYSTEM /EXEC UCX$BIND_SERVER000
$ DEASSIGN /SYSTEM /EXEC UCX$BIND_SERVER001
$ DEASSIGN /SYSTEM /EXEC UCX$BIND_DOMAIN
$ DEASSIGN /SYSTEM /EXEC TCPIP$BIND_SERVER000
$ DEASSIGN /SYSTEM /EXEC TCPIP$BIND_SERVER001
$ DEASSIGN /SYSTEM /EXEC TCPIP$BIND_DOMAIN
$ @IP$:START_SERVER
$ @IP$:START_SMTP
```

```
$
```

SET HOST-NAME

SET HOST-NAME — Specifies the local computer's host name and defines the default local domain.

Format

```
SET HOST-NAME host
```

Description

If your configuration includes Domain Name Service (DNS), you must specify the host name in dotted format, for example, `HAMLET.CALTECH.EDU` or `JETSON.SPROCKETS.COM`.

The default local domain is derived from the HOST-NAME parameter. For example, with a host name of `HAMLET.CALTECH.EDU`, the default local domain is `.CALTECH.EDU`. VSI TCP/IP uses the default local domain to complete abbreviated host names. For example, with the default of `CALTECH.EDU`, if you entered the command:

```
$ TELNET ROMEO
```

TELNET would attempt to connect to a `ROMEO.CALTECH.EDU` host.

You can change the value of HOST-NAME without rebooting by also defining or redefining the system-wide logical name `IP$HOST_NAME` and restarting the IP$SERVER (`@IP$:START_SERVER`) and `SMTP_SYMBIONT` (`@IP$START_STMP`) processes.

Parameter

host

Specifies the name of your host.

Example

This example sets the HOST-NAME to `HAMLET.CALTECH.EDU`, then issues the commands that change the parameter without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET HOST-NAME HAMLET.CALTECH.EDU
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXEC IP$HOST_NAME "HAMLET.CALTECH.EDU"
$ DEFINE /SYSTEM /EXEC ARPANET_HOST_NAME "HAMLET.CALTECH.EDU"
$ DEFINE /SYSTEM /EXEC UCX$INET_HOST "HAMLET.CALTECH.EDU"
$ DEFINE /SYSTEM /EXEC UCX$BIND_DOMAIN "FLOWERS.COM"
$ DEFINE /SYSTEM /EXEC TCPIP$INET_HOST "HAMLET.CALTECH.EDU"
$ DEFINE /SYSTEM /EXEC TCPIP$BIND_DOMAIN "FLOWERS.COM"
$ @IP$:START_SERVER
$ @IP$:START_SMTP
$
```

SET IP-CLUSTER-ALIASES

SET IP-CLUSTER-ALIASES — Specifies a list of cluster-wide Internet addresses to which this node should respond.

Format

`SET IP-CLUSTER-ALIASES` [*`address`* [*`address . . .`*]]

Description

In a OpenVMScluster, the nodes coordinate among themselves so that only one node responds to the cluster-wide addresses at any one time. Should that node fail or be shut down, another node immediately takes over this task.

You can use this parameter with connectionless protocols (such as NFS) for automatic failover. *`IP-CLUSTER-ALIASES`* specifies a list of IP addresses to which this node should respond in addition to the address(es) that are configured for the interfaces you define.

Disable *`IP-CLUSTER-ALIASES`* by entering the value without specifying an address.

You can change the value of *`IP-CLUSTER-ALIASES`* without rebooting by also defining or redefining the system-wide logical name `IP$IP_CLUSTER_ALIASES` and restarting the IP$SERVER (`@IP$:START_SERVER`) process.

Parameter

[*`address`* [*`address . . .`*]]

Specifies the Internet address to which to respond.

Example

This example sets *`IP-CLUSTER-ALIASES`* to 192.1.1.2, then executes the commands that change this parameter without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET IP-CLUSTER-ALIASES 192.1.1.2
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$IP_CLUSTER_ALIASES "192.1.1.2"
$ @IP$:START_SERVER
```

SET LOAD-PWIP-DRIVER

SET LOAD-PWIP-DRIVER — Enables automatic loading of the version 5 PATHWORKS server interface (PWIP) driver when VSI TCP/IP starts. You must enable the PWIP driver to use version 5 PATHWORKS. Earlier versions of PATHWORKS use the UCX interface. Reboot your system after setting LOAD-PWIP-DRIVER so that the change is recognized by the operating system. Formerly VMS/ULTRIX Connection (UCX). Referred to generically throughout the this documentation as UCX.

Format

`SET LOAD-PWIP-DRIVER` { TRUE | FALSE }

Parameters

TRUE

Enables the PWIP driver.

FALSE

Disables the PWIP driver (the default setting).

SET LOCAL-DOMAIN

SET LOCAL-DOMAIN — Overrides the default local domain derived from a previous **SET HOST-NAME** command. If your configuration includes a Domain Name System (DNS) domain server, you can use the *`LOCAL-DOMAIN`* parameter to override the default local domain derived from the *`HOST-NAME`* parameter. You can change the value of *`LOCAL-DOMAIN`* without rebooting by also defining or redefining the system-wide logical name `IP$LOCALDOMAIN` and restarting the IP$SERVER (`@IP$:START_SERVER`) and `SMTP_SYMBIONT` (`@IP$START_SMTP`) processes.

Format

`SET LOCAL-DOMAIN` *`domain`*

Parameter

`domain`

Specifies the default domain name.

Example

This example sets *`LOCAL-DOMAIN`* to `TREEFROG.COM`, then invokes the commands that change this parameter without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
```

```
NET-CONFIG>SET LOCAL-DOMAIN TREEFROG.COM
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$LOCALDOMAIN "TREEFROG.COM"-
$ DEFINE /SYSTEM /EXECUTIVE UCX$BIND_DOMAIN "TREEFROG.COM"
$ DEFINE /SYSTEM /EXECUTIVE TCPIP$BIND_DOMAIN "TREEFROG.COM"
$ @IP$:START_SERVER
$ @IP$:START_SMTP
$
```

SET LPD-DEFAULT-USERNAME

SET LPD-DEFAULT-USERNAME — Specifies a default user name for print jobs received from a remote system via the LPD protocol. The *`LPD-DEFAULT-USERNAME`* parameter allows all print jobs requested by remote users to be processed, regardless of whether each user has a local OpenVMS user name. The parameter defines a default user name for users without local user names. You can change the value of *`LPD-DEFAULT-USERNAME`* without rebooting by also defining or redefining the system-wide logical name `IP$LPD_DEFAULT_USERNAME`.

Format

```
SET LPD-DEFAULT-USERNAME user
```

Parameter

`user`

Specifies the default user name, which must be an existing name in the OpenVMS system.

Example

This example sets *`LPD-DEFAULT-USERNAME`* to PYWACKET, then executes the commands that change this parameter without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from
IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET LPDD-DEFAULT-USERNAME PYWACKET
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$FPD_DEFAULT_USERNAME "PYWACKET"
```

SET NAMESERVER-RETRANSMISSION

SET NAMESERVER-RETRANSMISSION — Controls the time between requests and the number of name server requests made before the system stops sending requests to a nonresponding server. The first argument is the time in seconds; the optional second argument is the number of tries to make. You can change the value of *NAMESERVER-RETRANSMISSION* without rebooting by also defining or redefining the system-wide logical names IP$NAMESERVER_RETRANS and IP$NAMESERVER_RETRY and restarting the IP$SERVER (@IP$:START_SERVER) and SMTP_SYMBIONT (@IP$START_SMTP) processes.

Format

SET NAMESERVER-RETRANSMISSION *seconds retries*

Parameters

seconds

Specifies the time between retransmissions, in seconds (by default, 4 seconds).

retries

Specifies the number of retransmissions to make before giving up (by default, 4 tries).

Example

This example specifies that the name server waits eight seconds between retries, and makes up to four retries before the system stops sending requests to the nonresponding server. The commands after setting this parameter allow the parameter to take effect immediately without rebooting the system.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET NAMESERVER-RETRANSMISSION 8 4
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$NAMESERVER_RETRANS 8
$ DEFINE /SYSTEM /EXECUTIVE IP$NAMESERVER_RETRY 4
$ @IP$:START_SERVER
$ @IP$:START_SMTP
```

SET SNMP-MAX-CONNECTIONS

SET SNMP-MAX-CONNECTIONS — Defines the maximum number of connections that the SNMP agent will report

Format

SET SNMP-MAX-CONNECTIONS *number*

Parameter

number

Specifies the maximum number of connections that he SNMP agent will report on.

Description

Use the *SNMP-MAX-CONNECTIONS* parameter to increase the maximum number of connections that the SNMP agent will report on. A number that is too low will prevent the SNMP agent from reporting on any connections. If this parameter is not set, the SNMP agent uses the default of 256 connections. The maximum value is 3276.

SET SNMP-MAX-ROUTES

SET SNMP-MAX-ROUTES — Defines the maximum number of routes that the SNMP agent will report

Format

```
SET SNMP-MAX-ROUTES number
```

Parameter

number

Specifies the maximum number of routes that he SNMP agent will report on.

Description

Use the *SNMP-MAX-ROUTES* paramter to increase the maximum number of routes that the SNMP agen will report on. A number that is too low will prevent the SNMP agent from reporting on any routess. If this parameter is not set, the the SNMP agent uses the default of 256 connections. The maximum value is 2978.

SET SPOOL-DIRECTORY

SET SPOOL-DIRECTORY — Changes the VSI TCP/IP spool directory from its default of `IP$COMMON_ROOT:[IP.SPOOL]`. The spool directory is used to store transient mail and print files. You can change the value of *SPOOL-DIRECTORY* without rebooting by also defining or redefining the system-wide logical name `IP$SPOOL`.

Format

```
SET SPOOL-DIRECTORY directory
```

Parameter

directory

Specifies the new spool directory.

Example

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
```

```
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET SPOOL-DIRECTORY SYS$SYSROOT:[TMP]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE IP$SPOOL "SYS$SYSROOT:[TMP]"
```

SET TFTP-DIRECTORY

SET TFTP-DIRECTORY — Defines the TFTP server's default directory.

Format

`SET TFTP-DIRECTORY` *`directory`*

Parameter

`directory`

Specifies the name of the new default directory.

Description

Use the *`TFTP-DIRECTORY`* parameter to define the TFTP server's default directory. When the parameter is not set, there is no default directory.

You can change the value of *`TFTP-DIRECTORY`* without rebooting by also defining or redefining the system-wide logical name `IP$TFTP_DEFAULT_DIRECTORY`, then issuing the **IP NETCONTROL TFTP RELOAD** command.

Example

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET TFTP-DIRECTORY USERS:[TFTP-FILES]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ DEFINE /SYSTEM /EXECUTIVE -
_$ IP_TFTP_DEFAULT_DIRECTORY "USERS:[TFTP-FILES]"
$ IP NETCONTROL TFTP RELOAD
$
```

SET TIMEZONE

SET TIMEZONE — Although OpenVMS does not keep track of timezones, VSI TCP/IP requires this information. The *`TIMEZONE`* parameter sets the timezone for your system and indirectly specifies the offset from GMT (and UTC) at which the local OpenVMS clock is run. VSI TCP/IP automatically adjusts for Daylight Savings Time (DST) if appropriate for the rules in effect. If your OpenVMS clock time and your local time differ, set the *`TIMEZONE`* parameter to correspond to the OpenVMS clock. You can change the value of *`TIMEZONE`* without rebooting by using the **IP SET /TIMEZONE** command.

Format

```
SET TIMEZONE timezone
```

Parameter

`timezone`

Specifies the abbreviation for the timezone, for example, PST.

Example

This example sets the timezone to PST (Pacific Standard Time), then uses **SET /TIMEZONE** to assign the same value to the system-wide variable.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET TIMEZONE PST
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ IP SET/TIMEZONE PST
$
```

SET TIMEZONE-RULES

SET TIMEZONE-RULES — Sets the timezone rules in effect for your system. The normal timezone rules in VSI TCP/IP include zones for the United States and a number of other countries around the world. Certain countries, such as Canada, have their own Daylight Savings Time rules for timezones such as PST, Pacific Standard Time. The *`TIMEZONE-RULES`* parameter is used to override the default rules. You can change the value of *`TIMEZONE-RULES`* without rebooting by using the **IP SET /TIMEZONE** command.

Format

```
SET TIMEZONE-RULES zonelist
```

Parameter

`zonelist`

Specifies a comma-separated list of countries and timezones for which to load rules. These can be found in `IP$:TIMEZONES.DAT`.

Example

This example sets the *`TIMEZONE-RULES`* to *`US/ARIZONA`*, then sets the system-wide variable to the same value so that the system does not have to be rebooted for the new rules to take effect.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET TIMEZONE-RULES US/ARIZONA
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$ IP SET/TIMEZONE MST/SELECT="US/ARIZONA"
$
```

SET WHOIS-DEFAULT-SERVER

SET WHOIS-DEFAULT-SERVER — Sets the default server for the **WHOIS** command. The **WHOIS** command displays host information obtained from the `RS.INTERNIC.NET` host server. By default, **WHOIS** connects to the server on the host `RS.INTERNIC.NET`, but can be overridden using the **/HOST** qualifier. Use this command to change the default server. You can change the value of *`WHOIS-DEFAULT-SERVER`* without rebooting by also defining or redefining the system-wide logical name `IP$WHOIS_DEFAULT_SERVER`.

Format

```
SET WHOIS-DEFAULT-SERVER host
```

Parameter

`host`

Specifies the name of the server to which to connect. If not specified, the host defaults to `RS.INTERNIC.NET`.

Example

This example sets *`WHOIS-DEFAULT-SERVER`* to `FNORD.FOO.COM`, then sets the system-wide logical name `IP$WHOIS_DEFAULT_SERVER` to the same value to avoid rebooting the system after changing the **WHOIS** default server parameter.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SET WHOIS-DEFAULT-SERVER FNORD.FOO.COM
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
```

```
$ DEFINE /SYSTEM /EXECUTIVE -
_$ IP_WHOIS_DEFAULT_SERVER "FNORD.FOO.COM"
```

SET WINS-COMPATIBILITY

SET WINS-COMPATIBILITY — Enables automatic definition of the logical names required for compatibility with applications developed for the WIN/TCP and Pathway products from The Wollongong Group. The logical names allow VSI TCP/IP to support applications that run under those products. You must also generate a UNIX-format host table using the **IP HOST_TABLE COMPILE/UNIX** command.

Format

`SET WINS-COMPATIBILITY` { TRUE | FALSE }

Parameters

TRUE

Enables the definition of the WIN/TCP and Pathway compatibility logical names.

FALSE

Disables the definition of the WIN/TCP and Pathway compatibility logical names.

Examples

1.
   ```
   $ IP CONFIGURE /INTERFACE
   VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
   [Reading in MAXIMUM configuration from IP$:IP.EXE]
   [Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
   NET-CONFIG>SET LOCAL-DOMAIN TREEFROG.COM
   NET-CONFIG>EXIT
   [Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
   [Writing Startup file IP$SYSTARTUP.COM]
   [Changes take effect after the next OpenVMS reboot]
   ```

2. Use the following DCL commands to enable WINS compatibility mode without a reboot:

   ```
   $ DIRECTORY = F$TRNLNM("IP$SPECIFIC_ROOT")  - "]"
   DEFINE/SYSTEM/EXEC/TRANSLATION_ATTRIBUTES=CONCEAL TWG$ETC -
   'DIRECTORY'IP.], 'DIRECTORY'SYSCOMMON.IP.]
   DEFINE/SYSTEM/EXEC/TRNASLATION_ATTRIBUTES=CONCEAL TWG$TCP -
   DEFINE/SYSTEM/EXEC INET_DOMAIN_NAME 'FLOWERS.COM"
   DEFINE/SYSTEM/EXEC INET_NAMESERVER_LIST "127.0.0.1"
   DEFINE/SYSTEM/EXEC WINS_MAILSHR IP$:SMTP_MAILSHR
   DEFINE/SYSTEM/EXEC DECW$TRANSPORT_WINTCP DECW$TRANSPORT_TCIP
   ```

3. Whether you choose to reboot or not, use the following commands to create a UNIX-format host table:

   ```
   $ SET DEFAULT IP$COMMON_ROOT:[IP]
   $ IP HOST_TABLE COMPILE /UNIX
   $
   ```

SHOW

SHOW — Displays the current VSI TCP/IP device configuration.

Format

```
SHOW command
```

Parameter

command

Specifies the type of display. Accepted values are *CURRENT* (the default) or *MAXIMUM*. If a command is not entered, the default is *CURRENT*.

Example

This example lists the current configuration.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SHOW
```

Interface		Adapter	CSR Address	Flags/Vector
se0)	(Shared OpenVMS Ethernet [TCP/IP$: 192.41.228.78] [VMS Device: ESA0]	-NONE-	-NONE-	-NONE-
s10	(Serial Line IP) [TCP/IP$: 192.41.228.78==>192.41.228.80] [VMS Terminal: TTA2, 9600 Baud] [Header Compression DISABLED]	-NONE-	-NONE-	-NONE-
*s11	(Serial Line IP) [VMS Terminal: TXA0] [Header Compression DISABLED]	-NONE-	-NONE-	-NONE-

```
Official Host Name:        BANANA.SLUG.COM
Default IP Route:          192.41.228.71
Domain Nameservers:        127.0.0.1
Local Domain:              slug.com
Timezone:                  PST
```

```
SMTP Host Name:            SLUG.COM
Default RMT Tape:          MKB100:
Default TFTP Directory:    IP$ROOT:[IP.TFTP]
Anonymous FTP Directory:   USERS:[ANONYMOUS]
Load UCX $QIO driver:      TRUE
WINS Compatibility:        TRUE
NET-CONFIG>QUIT
$
```

SPAWN

SPAWN — Executes a single DCL command, or if entered without options, starts a subprocess with the same effect as **PUSH**. To return from DCL, use the **LOGOUT** command. If the `IP$DISABLE_SPAWN` logical is set, **SPAWN** does not work.

Format

`SPAWN` [*`command`*]

Parameter

[*`command`*]

Specifies a command to execute. If you omit command, a DCL command line subprocess is created.

Qualifiers

`/INPUT=`*`file-spec`*

Specifies an input file to the command you enter with SPAWN.

`/LOGICAL_NAMES`

`/NOLOGICAL_NAMES`

Specifies that logical names and logical name tables are not copied to the subprocess.

`/SYMBOLS`

`/NOSYMBOLS`

Specifies that global and local names are not passed to the subprocess.

`/WAIT`

`/NOWAIT`

Returns control without waiting for the command to complete. Do not use this qualifier with commands that have prompts or screen displays.

`/OUTPUT=`*`file-spec`*

Specifies a file that retains the output of the command invoked with **SPAWN**. This qualifier only works when a single command is entered without creating a DCL subprocess. In addition, this qualifier is positional; you must enter it immediately after **SPAWN** or other qualifiers.

Examples

This example displays terminal information, captures the output in a file, then displays the information with the **TYPE** command.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SPAWN/OUTPUT=FOO. SHOW TERM
NET-CONFIG>SPAWN TYPE FOO.
...
```

This example invokes a command procedure.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SPAWN @COMPROC
...
```

This example displays help information about the NET-CONFIG utility. Use the **LOGOUT** command to return control to NET-CONFIG.

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SPAWN
$ HELP IP CONFIGURE /NETWORK_DEVICES
...
$ LOGOUT
NET-CONFIG>
```

STATUS

STATUS — Displays the status of the current configuration.

Format

```
STATUS
```

Example

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>STATUS
There is the VSI TCP/IP for OpenVMS network configuration program Version
 10.5 (nnn)
There are 3/1024 devices in the current configuration.
There are 190/1024 devices in the MAXIMUM configuration.
The configuration IP$:NETWORK_DEVICES.CONFIGURATION is not modified.
The startup file IP$SYSTARTUP.COM is not modified.
NET-CONFIG>QUIT
$
```

USE

USE — Reads in a configuration file. (Functionally equivalent to **GET**.)

Format

USE *config_file*

Parameter

config_file

Specifies the name of the configuration file to read in.

VERSION

VERSION — Displays the NET-CONFIG version and release information.

Format

VERSION

Example

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>VERSION
This is the VSI TCP/IP for OpenVMS network configuration program Version
 10.5 (nnn)
NET-CONFIG>QUIT
$
```

WRITE

WRITE — Writes the current configuration to a file. (Functionally equivalent to **SAVE**.)

Format

WRITE [*config_file*]

Parameter

[*config_file*]

Specifies the name of the configuration file to write out (by default, the same file from which the configuration was read).

Example

```
$ IP CONFIGURE /INTERFACE
VSI TCP/IP for OpenVMS Network Configuration Utility 10.5 (nnn)
[Reading in MAXIMUM configuration from IP$:IP.EXE]
[Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>WRITE
[Writing configuration to IP$ROOT:[IP]NETWORK_DEVICES.CONFIGURATION.7]
NET-CONFIG>EXIT
[Writing configuration to IP$:NETWORK_DEVICES.CONFIGURATION]
[Writing Startup file IP$SYSTARTUP.COM]
[Changes take effect after the next OpenVMS reboot]
$
```

Chapter 5. NFS-CONFIG Command Reference

For VSI TCP/IP V10.5, NFS-CONFIG commands are documented in the *VSI TCP/IP Administrator's Guide: Volume II, Chapter 13: Configuring the VSI TCP/IP NFS Client & Server*.

Chapter 6. NTYCP Command Reference

This chapter describes the Network Terminal Device Control Program (NTYCP) commands you can use to create terminal devices.

To invoke NTYCP as an OpenVMS "foreign" command:

```
$ NTYCP := $IP$:NTYCP
```

To invoke NTYCP interactively:

```
$ RUN IP$:NTYCP
NTYCP> CREATE PORT NTYnnnn /NODE=host-name /PORT=port-number
NTYCP> EXIT
```

To set up the terminal characteristics:

```
$ SET TERMINAL NTYnnn:/PERMANENT/NOBROADCAST/NOTYPEAHEAD/NOWRAP
```

To set up spooling:

```
$ SET DEVICE/SPOOLED=(queue-name,SYS$SYSDEVICE:) NTYnnnn:
```

To initialize and start the queue:

```
$ INITIALIZE/QUEUE/ON=NTYnnnn: queue-name/PROCESSOR=IP_NTYSMB/START
```

This example shows how to set up a print queue connected to an HP LaserJet printer with a JetDirect card:

```
$ NTYCP := $IP$:NTYCP
$ NTYCP CREATE PORT NTY1001/NODE=hp-laserjet/PORT=9100
%NTYCP-S-CREPORT, device _NTY1001: created to host 192.1.1.5, port 9100
$ SET TERMINAL/PERMANENT NTY1001:/NOBROADCAST/NOTYPEAHEAD/NOWRAP
$ INITIALIZE/QUEUE/ON=NTY1001: HP_LASERJET/PROCESSOR=IP_NTYSMB/START
```

Command Summary

Table 6.1 lists the NTYCP commands.

Table 6.1. NTYCP Command Summary

NTYCP Command	Description
CREATE PORT	Creates a new network terminal port device.
DELETE PORT	Deletes an NTY device created by the **CREATE PORT** command.
EXIT	Exits from NTYCP to DCL command mode.
HELP	Displays help text about NTYCP commands.
MODIFY PORT	Modifies (changes) an existing network terminal port device.

CREATE PORT

CREATE PORT — Creates a new network terminal port device. The device links the OpenVMS terminal driver to a TCP/IP network connection directed to the destination address specified by the /

HOST qualifier and the **/PORT** or **/SERVICE** qualifier. You can use network terminal (NTY) devices with the `IP$NTYSMB` print symbiont to provide OpenVMS print queue support for network-connect printers. You can also use them with user-written applications that need a simple terminal-style I/O interface to a remote terminal, plotter, etc.

Format

```
CREATE PORT device-name
```

Command Qualifiers	**Defaults**
/[NO]LOG /LOGICAL=(*logical-name-options...*) /NODE=*node-name-or-address* /PORT=*port-number* /SERVICE=*service-name*	/LOG

Parameter

device-name

Name of the NTY port device to be created. If specified, the device name must be of the form NTY*n*, where *n* is a device unit number, which must be in the range 1-9999. The specified device must not already exist. If omitted, the next available unit number will be used.

Qualifiers

/LOG

Controls whether a log message is generated on successful completion of the command. The default is **/LOG**.

/LOGICAL=(*logical-name-options...*)

Causes NTYCP to create a logical name for the created NTY device. This qualifier takes one or more of the keyword options specified in Table 6.2. If you specify multiple options, separate them by commas. You must have access to the specified logical name table and sufficient privilege to create the logical name.

Table 6.2. NTYCP CREATE PORT /LOGICAL Keyword Options

Keyword	**Description**
NAME=*logical-name*	The logical name to be created. You must specify this option if you use **/LOGICAL**.
TABLE=*table-name*	Specifies the logical name table in which the logical name should be created. This can be the actual name of a table, or one of the key-words PROCESS, GROUP, SYSTEM. The default is TABLE=PROCESS.

Keyword	Description
MODE=`mode-name`	Specifies the access mode in which the logical name should be created. The keywords are EXECUTIVE, SUPERVISOR, USER. The default is MODE=SUPERVISOR.

`/NODE=`*`node-name-or-address`*

Specifies the name or numeric IP address of the remote node. You must specify this qualifier.

If you specify a node name, it is translated into an IP address. Only one IP address may be configured per NTY device. If the destination system is a host with multiple IP addresses, and not all addresses are directly reachable from your local system, you should specify the IP address numerically to ensure that a reachable address is configured.

`/PORT=`*`port-number`*

Specifies a TCP port number on the remote node to which the connection will be made. You must specify either the /PORT qualifier or the **/SERVICE** qualifier with the command.

`/SERVICE=`*`service-name`*

Specifies the name of a TCP service that translates to a port number to which the connection will be made. You may specify any TCP service name present in the local hosts/services table. You must specify either the **/PORT** qualifier or the **/SERVICE** qualifier with the command.

DELETE PORT

DELETE PORT — Deletes an NTY device created by the NTYCP **CREATE PORT** command.

Format

```
DELETE PORT device-name
```

Command Qualifiers	Defaults
`/[NO]LOG`	`/LOG`

Parameter

`device-name`

Name of the NTY port device to be deleted.

Qualifier

`/LOG`

Controls whether a log message is generated on successful completion of the command. The default is **/LOG**.

EXIT

EXIT — Causes NTYCP to exit back to DCL command mode.

Format

```
EXIT
```

HELP

HELP — Displays help text about NTYCP commands.

Format

HELP [*topic*]

Parameter

[*topic*]

A command name or other topic in the NTYCP help library. If omitted, a list of topics displays.

MODIFY PORT

MODIFY PORT — Modifies (changes) an existing network terminal port device. The device links the OpenVMS terminal driver to a TCP/IP network connection directed to the destination address specified by the **/HOST** qualifier and the **/PORT** or **/SERVICE** qualifier.

Format

MODIFY_PORT *device-name*

Command Qualifiers	**Defaults**
/[NO]LOG /LOGICAL=(*logical-name-options...*) /NODE=*node-name-or-address* /PORT=*port-number* /SERVICE=*service-name*	/LOG

Parameters

modify-object

Name of the NTY object to be modified (changed).

port-name

Name of the NTY port device to be modified (changed).

Qualifiers

`/LOG`

Controls whether a log message is generated on successful completion of the command. The default is **/LOG**.

`/LOGICAL=(`*`logical-name-options...`*`)`

This is the logical name for the NTY device you want to modify. This qualifier takes one or more of the keyword options specified in Table 6.3. If you specify multiple options, separate them by commas. You must have access to the specified logical name table and sufficient privilege to modify the logical name.

Table 6.3. NTYCP MODIFY PORT /LOGICAL Keyword Options

Keyword	Description
NAME=*logical-name*	The logical name to be created. You must specify this option if you use **/LOGICAL**.
TABLE=*table-name*	Specifies the logical name table in which the logical name should be created. This can be the actual name of a table, or one of the keywords PROCESS, GROUP, SYSTEM. The default is TABLE=PROCESS.
MODE=*mode-name*	Specifies the access mode in which the logical name should be created. The keywords are EXECUTIVE, SUPERVISOR, USER. The default is MODE=SUPERVISOR.

`/NODE=`*`node-name-or-address`*

Specifies the name or numeric IP address of the remote node. You must specify this qualifier.

If you specify a node name, it is translated into an IP address. Only one IP address may be configured per NTY device. If the destination system is a host with multiple IP addresses, and not all addresses are directly reachable from your local system, you should specify the IP address numerically to ensure that a reachable address is configured.

`/PORT=`*`port-number`*

Specifies a TCP port number on the remote node to which the connection will be made. You must specify either the /PORT qualifier or the **/SERVICE** qualifier with the command.

`/SERVICE=`*`service-name`*

Specifies the name of a TCP service that translates to a port number to which the connection will be made. You may specify any TCP service name present in the local hosts/services table. You must specify either the **/PORT** qualifier or the **/SERVICE** qualifier with the command.

Chapter 7. PRINTER-CONFIG Command Reference

This chapter describes the commands you can run from the PRINTER-CONFIG command line. With PRINTER-CONFIG you can examine, modify, and save configuration files for VSI TCP/IP remote print queues.

To invoke PRINTER-CONFIG:

```
$ IP CONFIGURE /PRINTERS
```

At the PRINTER-CONFIG prompt, type ? to list the available commands. For online help use the PRINTER-CONFIG **HELP** command.

Changes do not take effect until you do one of the following:

- Restart the VSI TCP/IP remote printer queues with the @IP$:REMOTE-PRINTER-QUEUES.COM command.
- Restart your system.

For details on configuring VSI TCP/IP remote printer queues, refer to the *VSI TCP/IP Administrator's Guide: Volume II*.

Command Summary

Table 7.1 lists the commands you can run from the PRINTER-CONFIG prompt.

Table 7.1. PRINTER-CONFIG Command Summary

PRINTER-CONFIG Command	Description
ADD	Adds a new OpenVMS print queue to the current configuration.
ATTACH	Switches the terminal to another process.
CLEAR	Removes all printer queues from the current configuration.
DELETE	Removes a single printer queue from the current configuration.
ERASE	Removes all printer queues from the current configuration (same as **CLEAR**).
EXIT	Saves the current printer configuration and leaves PRINTER-CONFIG mode.
GET	Reads in a printer configuration file.
HELP	Displays information about one or all commands.
MODIFY	Changes a printer configuration file.
PUSH	Accesses the DCL command line and pauses PRINTER-CONFIG.
QUIT	Exits PRINTER-CONFIG and prompts to save changes.
SAVE	Writes out the current printer configuration file (same as **WRITE**).
SELECT	Picks the printer that will be modified by subsequent SET commands.

PRINTER-CONFIG Command	**Description**
SET ALLOW-USER-SPECIFIED-PRINTER	Controls whether the print queue allows the use of **PRINT/ PARAMETER=(...)** for specifying the destination address or printer for an LPD job (to override the original queue configuration).
SET BASE-PRIORITY	Specifies the base process priority at which jobs are initiated from a batch execution queue.
SET BLOCK-LIMIT-LOWER	Limits the size of print jobs that can be processed on an output execution queue.
SET BLOCK-LIMIT-UPPER	Limits the size of print jobs that can be processed on an output execution queue.
SET BURST	Controls whether two file flag pages with a burst bar between them are printed preceding output.
SET CHARACTERISTICS	Specifies one or more characteristics for processing jobs on an execution queue.
SET DEFAULT-FORM	Sets the default form used when submitting a print job to this printer.
SET DESCRIPTION	Specifies a string of up to 255 characters used to provide operator-supplied information about the queue.
SET FLAG	Forces a OpenVMS banner page to print at the beginning of each file, by default, on the print queue.
SET LIBRARY	Sets the device control library for the print queue.
SET NOFEED	Prevents the OpenVMS print symbiont formatting code from inserting a form feed between pages.
SET OWNER	Sets the owner of the print queue.
SET PROTECTION	Sets the protection of the print queue.
SET RETAIN-ON-ERROR	Retains jobs that terminate in an error in the queue.
SET SCHEDULE-NOSIZE	Prints jobs in the order they were submitted, regardless of size.
SET SEPARATE-BURST	Specifies whether two job flag pages with a burst bar between them are printed at the beginning of each job.
SET SEPARATE-FLAG	Specifies whether a job flag page is printed at the beginning of each job.
SET SEPARATE-RESET	Specifies one or more device control library modules that contain the job reset sequence for the queue.
SET SEPARATE-TRAILER	Specifies whether a job flag page is printed at the end of each job.
SET SUPPRESS-EOJ-FF	When set on a STREAM queue, the OpenVMS print symbiont formatting code does not add a form feed to the end of the job.
SET SUPPRESS-REMOTE-BANNER	When set on an LPD queue, the remote LPD is informed not to print a banner page; many LPD servers do not support this option.
SET SUPPRESS-TELNET	When set on a STREAM queue, VSI TCP/IP does not try to use the TELNET protocol to communicate with the printer.
SET TAB-EXPAND	Forces the OpenVMS print symbiont formatting code to expand **TAB** characters into the correct number of **SPACE** characters.

PRINTER-CONFIG Command	Description
SET TRAILER	Controls whether a file trailer page is printed following output.
SET WS-DEFAULT	Defines for a batch job a working set default, the default number of physical pages that the job can use.
SET WS-EXTENT	Defines for the batch job a working set extent, the maximum amount of physical memory that the job can use.
SET WS-QUOTA	Defines for a batch job a working set quota, the amount of physical memory that is guaranteed to the job.
SHOW	Displays the current printer configuration.
SPAWN	Invokes a DCL command in PRINTER-CONFIG, or starts a subprocess.
STATUS	Displays the status of the printer configuration.
USE	Reads in a configuration file (same as GET).
VERSION	Displays the PRINTER-CONFIG version and release information.
WRITE	Writes out the current printer configuration file.

ADD

ADD — Adds a new OpenVMS print queue to the current VSI TCP/IP configuration, and prompts for queue configuration parameters.

Format

`ADD` *`queue_name`*

Parameter

`queue_name`

Specifies the name of the queue to add to the configuration.

Examples

This example adds a remote printer queue that prints via LPD on remote system 192.0.0.15.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility V10.5(nn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>ADD SYS$LASER
[Adding new configuration entry for queue "SYS$LASER"]
Remote Host Name or IP address: 192.0.0.15
Protocol Type: [LPD] LPD
Remote Queue Name: [lp]: lp
[SYS$LASER => 192.0.0.15, lp]
PRINTER-CONFIG>
```

This example adds a remote printer queue that prints data by connecting to TCP port 1395 at address 192.0.0.98.

```
PRINTER-CONFIG>ADD SYS$LPTERM
[Adding new configuration entry for queue “SYS$LPTERM”]
Remote Host Name or IP address: 192.0.0.98
Protocol Type: [LPD] STREAM
TCP Port Number: [23] 1395
[SYS$LPTERM => 192.0.0.98, TCP port 1395 (no telnet option negotiation)]
PRINTER-CONFIG>
```

Note

The Remote Queue Name specified may be case-sensitive. In particular, if the server is a UNIX system, you must specify it in the same case as it occurs in the UNIX /etc/printcap file, usually lowercase.

If the server is an Ethernet card in a printer, the name is not arbitrary. Check the Ethernet card documentation for the correct remote queue name.

ATTACH

ATTACH — Detaches the terminal from the calling process and reattaches it to another process. Use the **SPAWN SHOW PROCESS /SUBPROCESSES** command to list the names of subprocesses. Use the DCL **LOGOUT** command to return to the original process. **ATTACH** does not work if the `IP $DISABLE_SPAWN` logical is enabled.

Format

Attach *process-name*

Parameter

process-name

Specifies the name of a process to which you want your terminal attached. (Not all subprocesses can be attached; some testing may be required.)

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SPAWN
$ MM
MM>SPAWN SHOW PROCESS /SUB
...
There are 3 processes in this job:
_TWA42:
 PROC_1
  PROC_2  (*)
MM>ATTACH  _TWA42:
PRINTER-CONFIG>ATTACH PROC_1
MM>QUIT
$ LOGOUT
PRINTER-CONFIG>
```

This example shows the use and exit of attached subprocesses.

1. The first command uses SPAWN to create a subprocess. MM is invoked from the DCL command line. Next, the **SPAWN SHOW PROCESS /SUB** command is used to list all the subprocess names. The display shows that three subprocesses are active. (Process `_TWA42: is PRINTER-CONFIG`, `PROC_1` is MM, and `PROC_2` is the **SPAWN SHOW PROCESS /SUB** command.)

2. The MM>**ATTACH** command returns control to the PRINTER-CONFIG process. From this utility, **ATTACH** returns control to MM. To exit, **QUIT** is invoked from MM, and **LOGOUT** is invoked at the original spawned DCL command line; control returns back to PRINTER-CONFIG. (If **SPAWN SHOW PROCESS /SUB** had been entered, only this command and the configuration processes would be active.)

CLEAR

CLEAR — Removes all remote printer queues from the current VSI TCP/IP configuration.

Format

```
CLEAR
```

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>CLEAR
```

DELETE

DELETE — Removes the specified remote print queue from the current VSI TCP/IP configuration.

Format

```
DELETE queue_name
```

Parameter

queue_name

Specifies the name of the remote print queue to remove.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>DELETE SYS$LASER
```

ERASE

ERASE — Removes all printer queues from the current VSI TCP/IP configuration. (Functionally equivalent to **CLEAR**.)

Format

```
ERASE
```

EXIT

EXIT — Saves the current configuration, if it has been modified, then quits.

Format

```
EXIT
```

Examples

When the configuration has not changed, a message displays indicating that the configuration file is not updated.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>EXIT
[Configuration not modified, so no update needed]
$
```

When the configuration has been changed, a message displays indicating that the configuration file is updated.

```
PRINTER-CONFIG>EXIT
[Writing configuration to IP$:REMOTE-PRINTER-QUEUES.COM]
$
```

GET

GET — Reads in a VSI TCP/IP remote printer configuration file that defaults to `IP$:REMOTE-PRINTER-QUEUES.COM`. After a **GET**, you can use the various configuration commands to modify the printer configuration.

Format

GET *config-file*

Parameter

config-file

Specifies the name of the configuration file to read in.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>GET
[Reading in configuration from IP$ROOT:[IP]REMOTE-PRINTER-QUEUES.COM.68]
```

HELP

HELP — Invokes the command help.

Format

`HELP` [*topics*]

Parameter

[*topics*]

Contains a space-delimited list of topics that begins with a topic followed by subtopics. The default topic is **HELP**.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>HELP ?
ADD       ATTACH   CLEAR   DELETE   ERASE   EXIT    GET      HELP
MODIFY    PUSH     QUIT    SAVE     SHOW    SPAWN   STATUS   USE
VERSION   WRITE
PRINTER-CONFIG>
```

MODIFY

MODIFY — Changes the parameters of the specified queue in the VSI TCP/IP remote printer configuration.

Format

`MODIFY` *queue_name*

Parameter

queue_name

Specifies the name of the queue whose parameters you want to change.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>MODIFY REMOTE_LASER
[Modifying configuration entry for queue "REMOTE_LASER"]
Remote Host Name: [192.0.0.1] 192.0.0.2
Protocol Type: [LPD]
Remote Queue Name: [LASER]
[REMOTE_LASER => 192.0.0.2, LASER]
PRINTER-CONFIG>
```

PUSH

PUSH — Starts and attaches a DCL subprocess. If a parent process exists, attach to it. To return from DCL, use the **ATTACH** or the **LOGOUT** command. To switch back from a DCL subprocess, use the **ATTACH** command. **PUSH** does not work if the `IP$DISABLE_SPAWN` logical is set.

Format

```
PUSH
```

QUIT

QUIT — If the configuration file has been edited, **QUIT** prompts you to save the file before leaving.

Format

```
QUIT
```

SAVE

SAVE — Functionally equivalent to **WRITE**.

Format

```
SAVE
```

SELECT

SELECT — Picks the printer that will be modified by any subsequent **SET** commands.

Format

SELECT *printer*

Parameter

printer

Specifies the name of the printer to pick for modification.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>
```

SET ALLOW-USER-SPECIFIED-PRINTER

SET ALLOW-USER-SPECIFIED-PRINTER — Controls whether the print queue allows the use of **PRINT/PARAMETER=(...)** for specifying the destination address and/or printer for an LPD job (to override the original queue configuration).

Format

```
SET ALLOW-USER-SPECIFIED-PRINTER { enable | disable }
```

Parameter

{ *enable* | *disable* }

Specifies whether this function is enabled or disabled.

Example

This example shows how to enable the use of **PRINT/PARAMETER=(...)** to override the original queue configuration.

```
$ IP CONFIGURE /PRINTER
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility V10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SEL HP-PRINT
[The Selected Printer is now HP-PRINT]
PRINTER-CONFIG>SET ALLOW-USER-SPECIFIED-PRINTER ENABLE
```

SET BASE-PRIORITY

SET BASE-PRIORITY — Establishes the base priority of the symbiont process when the symbiont process is created. By default, if you omit this, the symbiont process is initiated at the same priority as the base priority established by DEFPRI at system generation (usually 4).

Format

```
SET BASE-PRIORITY priority
```

Parameter

priority

Specifies the base priority in decimal format, 0 to 15.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET BASE-PRIORITY 4
PRINTER-CONFIG>
```

SET BLOCK-LIMIT-LOWER

SET BLOCK-LIMIT-LOWER — Limits the size of print jobs that can be processed on the queue. Allows you to reserve certain printers for certain size jobs. You can set the lower block limit only if the upper block limit is also set (see **SET BLOCK-LIMIT-UPPER**).

Format

SET BLOCK-LIMIT-LOWER *lowlim*

Parameter

lowlim

The lowlim parameter is a decimal number referring to the minimum number of blocks accepted by the queue for a print job. If a print job is submitted that contains fewer blocks than the lowlim value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is decreased sufficiently, the job is processed.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET BLOCK-LIMIT-LOWER 25
PRINTER-CONFIG>
```

SET BLOCK-LIMIT-UPPER

SET BLOCK-LIMIT-UPPER — Limits the size of print jobs that can be processed on the queue. Allows you to reserve certain printers for certain size jobs.

Format

SET BLOCK-LIMIT-UPPER *uplim*

Parameter

uplim

The uplim parameter is a decimal number referring to the maximum number of blocks that the queue accepts for a print job. If a print job is submitted that exceeds this value, the job remains pending until the block limit for the queue is changed. After the block limit for the queue is increased sufficiently, the job is processed.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
```

```
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET BLOCK-LIMIT-UPPER 300
PRINTER-CONFIG>
```

SET BURST

SET BURST — Controls whether two file flag pages with a burst bar between them are printed preceding output.

Format

`SET BURST` *`keyword`*

Parameter

`keyword`

If you specify the keyword...	Description
ALL (default)	These flag pages are printed before each fill in the job.
ONE	These flag pages are printed once before the first file in the job.
NONE	No flag pages are printed. It is equivalent to NOBURST.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET BURST ONE
PRINTER-CONFIG>
```

SET CHARACTERISTICS

SET CHARACTERISTICS — Specifies one or more characteristics for processing print jobs. If a queue does not have all the characteristics that have been specified for a job, the job remains pending. Only the characteristics specified are established for the queue.

Format

`SET CHARACTERISTICS` *`characteristic,...`*

Parameter

`characteristics,...`

Queue characteristics are installation specific. The characteristic parameter can be either a value from 0 to 127 or a characteristic name that has been defined by the **DEFINE /CHARACTERISTIC** command.

Parenthesis are not required; they are added automatically.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET CHARACTERISTICS 56
PRINTER-CONFIG>
```

SET DEFAULT-FORM

SET DEFAULT-FORM — Specifies the default form used when submitting a print job to the printer.

Format

`SET DEFAULT-FORM` *`formname`*

Parameter

`formname`

Specifies the name of a form previously defined on the system with the **DEFINE /FORM** command.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET DEFAULT-FORM POSTSCRIPT
[Default Form POSTSCRIPT]
PRINTER-CONFIG>
```

SET DESCRIPTION

SET DESCRIPTION — Specifies a string of up to 255 characters used to provide operator-supplied information about the queue.

Format

`SET DESCRIPTION` *`string`*

Parameter

`string`

Sequence of any printable characters, including spaces. Case of input is preserved. The string may optionally be enclosed in quotation marks (" ").

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET DESCRIPTION THIS IS A TEST QUEUE.
PRINTER-CONFIG>
```

SET FLAG

SET FLAG — By default, forces a OpenVMS banner page to print at the beginning of each file on the print queue.

Format

`SET FLAG` *`mode`*

Parameter

`mode`

If mode is ENABLE, banner pages are printed; if DISABLE, banner pages are not printed.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET FLAG ENABLE
PRINTER-CONFIG>
```

SET LIBRARY

SET LIBRARY — Sets the print queue's device control library.

Format

`SET LIBRARY` *`libraryfile`*

Parameter

`libraryfile`

Specifies the name of a text library located in SYS$LIBRARY to be used as the device control library for the print queue.

Example

```
$ IP CONFIGURE /PRINTERS
```

```
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET LIBRARY SYSDEVCTL
[LIBRARY SYSDEVCTL]
PRINTER-CONFIG>
```

SET NOFEED

SET NOFEED — Prevents the OpenVMS print symbiont formatting code from inserting a form feed between pages.

Format

SET NOFEED *mode*

Parameter

mode

If mode is ENABLE, form feeds are not inserted; if DISABLE, they are inserted.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET NOFEED ENABLE
PRINTER-CONFIG>
```

SET OWNER

SET OWNER — Sets the owner of the print queue.

Format

SET OWNER *owner*

Parameter

owner

Specifies the identifier or UIC of a user on the system.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
```

```
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET OWNER OPERATOR
[OWNER OPERATOR]
PRINTER-CONFIG>
```

SET PROTECTION

SET PROTECTION — Sets the protection of the print queue.

Format

SET PROTECTION *protection_string*

Parameter

protection_string

Specifies a OpenVMS queue protection mask.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET PROTECTION (S:RWED,O:REW,G:RE,W:RE)
[PROTECTION (S:RWED,O:REW,G:RE,W:RE)]
PRINTER-CONFIG>
```

SET RETAIN-ON-ERROR

SET RETAIN-ON-ERROR — Retains jobs in the queue that terminate in an error.

Format

SET RETAIN-ON-ERROR *mode*

Parameter

mode

If mode is ENABLE, jobs are retained; if DISABLE, jobs are not retained.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
```

```
PRINTER-CONFIG>SET RETAIN-ON-ERROR ENABLE
PRINTER-CONFIG>
```

SET SCHEDULE-NOSIZE

SET SCHEDULE-NOSIZE — Prints jobs in the order they were submitted, regardless of size.

Format

SET SCHEDULE-NOSIZE *mode*

Parameter

mode

If mode is ENABLE, jobs will print in the order they are submitted; if mode is DISABLE, jobs will print in order by size (shorter prints before longer).

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SCHEDULE-NOSIZE ENABLE
PRINTER-CONFIG>
```

SET SEPARATE-BURST

SET SEPARATE-BURST — Specifies whether two job flag pages with a burst bar between them are printed at the beginning of each job.

Format

SET SEPARATE-BURST *mode*

Parameter

mode

If mode is ENABLE, prints the flag pages; if the mode is DISABLE, will not print the flag pages.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SEPARATE-BURST ENABLE
```

```
PRINTER-CONFIG>
```

SET SEPARATE-FLAG

SET SEPARATE-FLAG — Specifies whether a job flag page is printed at the beginning of each job.

Format

SET SEPARATE-FLAG *mode*

Parameter

mode

If mode is ENABLE, job flag page will print; if it is DISABLE, job flag page will not print.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SEPARATE-FLAG ENABLE
PRINTER-CONFIG>
```

SET SEPARATE-RESET

SET SEPARATE-RESET — Specifies one or more device control library modules that contain the job reset sequence for the queue. The specified modules from the queue's device control library (by default SYS$LIBRARY:SYSDEVCTL) are used to reset the device at the end of each job. The RESET sequence occurs after any file trailer and before any job trailer. Thus, all job separation pages are printed when the device is in its RESET state.

Format

SET SEPARATE-RESET *module,...*

Parameter

module

This is the name of the device control library module.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
```

```
PRINTER-CONFIG>SET SEPARATE-RESET cosmos
PRINTER-CONFIG>
```

SET SEPARATE-TRAILER

SET SEPARATE-TRAILER — Specifies whether a job flag page is printed at the end of each job.

Format

```
SET SEPARATE-TRAILER mode
```

Parameter

mode

If mode is ENABLE, job flag page will print; if it is DISABLE, job flag page will not print.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SEPARATE-TRAILER ENABLE
PRINTER-CONFIG>
```

SET SUPPRESS-EOJ-FF

SET SUPPRESS-EOJ-FF — When set on a STREAM queue, the OpenVMS print symbiont formatting code does not add a form feed to the end of the job.

Format

```
SET SUPPRESS-EOJ-FF mode
```

Parameter

mode

If mode is ENABLE, a form feed is not inserted at the end of each job; if DISABLE, a form feed is inserted at the end of each job.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5(nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SUPPRESS-EOJ-FF ENABLE
```

```
PRINTER-CONFIG>
```

SET SUPPRESS-REMOTE-BANNER

SET SUPPRESS-REMOTE-BANNER — When set on an LPD queue, the remote LPD does not print a banner page. (Many LPD servers do not support this option.)

Format

```
SET SUPPRESS-REMOTE-BANNER mode
```

Parameter

mode

If mode is ENABLE, banner pages may or may not be generated on the remote system; if DISABLE, banner pages are generated on the remote system.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SUPPRESS-REMOTE-BANNER ENABLE
PRINTER-CONFIG>
```

SET SUPPRESS-TELNET

SET SUPPRESS-TELNET — When set on a STREAM queue, VSI TCP/IP does not try to use the TELNET protocol to negotiate options with the remote printer. Most terminal servers expect VSI TCP/IP to negotiate TELNET options, and most printers that connect directly to an IP network expect VSI TCP/IP not to do so.

Format

```
SET SUPPRESS-TELNET mode
```

Parameter

mode

If mode is ENABLE, TELNET options are not negotiated; if DISABLE, TELNET options are negotiated.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
```

```
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET SUPPRESS-TELNET ENABLE
PRINTER-CONFIG>
```

SET TAB-EXPAND

SET TAB-EXPAND — Forces the OpenVMS print symbiont formatting code to expand **TAB** characters into the correct number of **SPACE** characters.

Format

```
SET TAB-EXPAND mode
```

Parameter

mode

If mode is ENABLE, tabs are converted to **SPACES**; if DISABLE, tabs are not changed.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET EXPAND-TAB ENABLE
PRINTER-CONFIG>
```

SET TRAILER

SET TRAILER — Controls whether a file trailer page is printed following output.

Format

```
SET TRAILER keyword
```

Parameter

keyword

If you specify the keyword...	Description
ALL (default)	These flag pages are printed before each fill in the job.
ONE	These flag pages are printed once before the first file in the job.
NONE	No flag pages are printed. It is equivalent to *NOBURST*.

Example

```
$ IP CONFIGURE /PRINTERS
```

```
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET TRAILER ALL
PRINTER-CONFIG>
```

SET WS-DEFAULT

SET WS-DEFAULT — Establishes the working set default of the symbiont process for the queue when the symbiont process is created. The value set by this command overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Format

SET WS-DEFAULT *quota*

Parameter

quota

Specify the value as a number of 512-byte pagelets on Alpha systems.

Note

OpenVMS rounds this value up to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha. For further information, see the *OpenVMS System Manager's Manual*.

If you specify 0 or NONE, the working set default value defaults to the value specified in the UAF or by the SUBMIT command (if it includes a `WSDEFAULT` value).

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET WS-DEFAULT 27
PRINTER-CONFIG>
```

SET WS-EXTENT

SET WS-EXTENT — Establishes the working set extent of the symbiont process for the queue when the symbiont process is created. The value set by this command overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Format

SET WS-EXTENT *quota*

Parameter

`quota`

Specify the value as a number of 512-byte pagelets on Alpha.

Note

OpenVMS rounds this value up to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on Alpha.

If you specify 0 or NONE, the working set extent value defaults to the value specified in the UAF or by the SUBMIT command (if it includes a `WSEXTENT` value).

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET WS-EXTENT 0
PRINTER-CONFIG>
```

SET WS-QUOTA

SET WS-QUOTA — Establishes the working set quota of the symbiont process for the queue when the symbiont process is created. The value set by this command overrides the value defined in the user authorization file (UAF) of any user submitting a job to the queue.

Format

`SET WS-QUOTA` *`quota`*

Parameter

`quota`

Specify the value as a number of 512-byte pagelets on OpenVMS Alpha. OpenVMS rounds this value up to the nearest CPU-specific page so that actual amount of physical memory allowed may be larger than the specified amount on OpenVMS Alpha. For further information, see the *OpenVMS System Manager's Manual*.

If you specify 0 or NONE, the working set quota value defaults to the value specified in the UAF or by the SUBMIT command (if it includes a WSQUOTA value).

Working set default, working set quota, and working set extent values are included in each user record in the system UAF. You can specify working set values for individual jobs or for all jobs in a given queue. The decision table shows the action taken for different combinations of specifications that involve working set values.

Is the SUBMIT command value specified?	Is the queue value specified?	Action taken
No	No	Use the UAF value.
No	Yes	Use value for the queue.
Yes	Yes	Use smaller of the two values.
Yes	No	Compare specified value with UAF value; use the smaller.

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:REMOTE-PRINTER-QUEUES.COM]
PRINTER-CONFIG>SELECT TEST
[The Selected Printer is now TEST]
PRINTER-CONFIG>SET WS-QUOTA 12
PRINTER-CONFIG>
```

SHOW

SHOW — Displays the current VSI TCP/IP printer configuration.

Format

```
SHOW
```

Examples

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SHOW
Queue Name   IP Destination   Remote Queue Name
----------   --------------   -----------------
SYS$LASER    192.0.0.15       laser
SYS$LPTERM   192.0.0.98       TCP port 1395
PRINTER-CONFIG>
```

This example shows detailed queue characteristics for a specific printer called HP5.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SHOW HP5
Queue Name   IP Destination   Remote Queue Name
----------   --------------   -----------------
HP5          192.0.0.9        TCP PORT 9100
Device Control Library = HPLF3SI
Queue Owner = [SUPPORT,*]
Default Form = WHITEPAPER
End of Job Form Feed will be suppressed
Telnet Options Processing will be suppressed
PRINTER-CONFIG>
```

SPAWN

SPAWN — Executes a single DCL command, or if entered without options, starts a subprocess with the same effect as **PUSH**. To return from DCL, use the **LOGOUT** command. **SPAWN** does not work if the `IP$DISABLE_SPAWN` logical is set.

Format

`SPAWN` [*`command`*]

Parameter

[*`command`*]

Specifies a command to execute. If you omit command, a DCL command line subprocess is created.

Qualifiers

`/INPUT=`*`file-spec`*

Specifies an input file to the command you enter with **SPAWN**.

`/LOGICAL_NAMES`

`/NOLOGICAL_NAMES`

Specifies that logical names and logical name tables are not copied to the subprocess.

`/SYMBOLS`

`/NOSYMBOLS`

Specifies that global and local names are not passed to the subprocess.

`/WAIT`

`/NOWAIT`

Returns control without waiting for the command to complete. Do not use this qualifier with commands that have prompts or screen displays.

`/OUTPUT=`*`file-spec`*

Specifies a file that retains the output of the command invoked with **SPAWN**. This qualifier only works when a single command is entered without creating a DCL subprocess. In addition, this qualifier is positional; you must enter it immediately after **SPAWN** or other qualifiers.

Examples

This example displays terminal information, captures the output in a file, then displays the information with the **TYPE** command.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SPAWN/OUTPUT=FOO. SHOW TERM
```

```
PRINTER-CONFIG>SPAWN TYPE FOO.
...
```

This example invokes a command procedure.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SPAWN @COMPROC
...
```

This example displays help information about the PRINTER-CONFIG utility. Use the **LOGOUT** command to return control to PRINTER-CONFIG.

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>SPAWN
$ HELP IP CONFIGURE /PRINTERS
...
$ LOGOUT
PRINTER-CONFIG>
```

STATUS

STATUS — Shows the status of the VSI TCP/IP remote printer configuration program.

Format

```
STATUS
```

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5 (nnn)
PRINTER-CONFIG>STATUS
This is the VSI TCP/IP for OpenVMS Remote Printer configuration program
 Version Example
There are 1/1000 queues in the current configuration.
The configuration IP$:REMOTE-PRINTER-QUEUES.COM is not modified.
PRINTER-CONFIG>
```

USE

USE — Functionally equivalent to **GET**.

Format

```
USE
```

VERSION

VERSION — Shows the version and release information of the VSI TCP/IP remote printer configuration program.

Format

```
VERSION
```

Example

```
$ IP CONFIGURE /PRINTERS
VSI TCP/IP for OpenVMS Remote Printer Configuration Utility 10.5(nnn)
PRINTER-CONFIG>VERSION
This is the VSI TCP/IP for OpenVMS Remote Printer configuration program
 Version 10.5(nnn)
PRINTER-CONFIG>
```

WRITE

WRITE — Writes out the current VSI TCP/IP remote printer configuration to a VSI TCP/IP remote printer configuration file.

Format

```
WRITE config_file
```

Parameter

config_file

Specifies the name of the file to which to write the current VSI TCP/IP printer configuration (by default, the same file from which the configuration was read).

Chapter 8. SERVER-CONFIG Command Reference

This chapter describes the commands you can run from the SERVER-CONFIG command line. SERVER-CONFIG lets you examine, modify, and save configuration files for VSI TCP/IP services.

To invoke SERVER-CONFIG:

```
$ IP CONFIGURE /SERVERS
```

SERVER-CONFIG commands affect the configuration of the currently selected service. You can select services with the **SELECT** command. By default, no service is selected.

At any SERVER-CONFIG prompt, type ? to list the available commands. Use the SERVER-CONFIG **HELP** command to view online help for each SERVER-CONFIG command.

Changes do not take effect until you do one of the following:

- Reload and restart the VSI TCP/IP server process with the **IP NETCONTROL** command.
- Restart your system.

For details on configuring VSI TCP/IP services, refer to the *VSI TCP/IP Administrator's Guide: Volume II*.

Command Summary

Table 8.1 lists the commands you can run from the SERVER-CONFIG prompt.

Table 8.1. SERVER-CONFIG Command Summary

SERVER-CONFIG Command	Description
ADD	Adds a service to the current configuration.
ATTACH	Switches terminal control to another process.
COPY	Copies a service entry to the current configuration.
DELETE	Deletes a service from the current configuration.
DISABLE	Disables a service in the current configuration.
ENABLE	Enables a service in the current configuration.
EXIT	Exits from the SERVER-CONFIG session.
GET	Reads a server configuration file; same as **GET**.
HELP	Displays command information.
NETCONTROL	Contacts the NETCONTROL server at another site.
PUSH	Accesses the DCL command line while pausing SERVER-CONFIG.
QUIT	Exits SERVER-CONFIG and prompts to save changes.
RESTART	Restarts the master server process.
SAVE	Writes out the current server configuration file.
SELECT	Selects a server for **SET** commands.

SERVER-CONFIG Command	**Description**
SET ACCEPT-HOSTS	Specifies which hosts can access the server.
SET ACCEPT-NETS	Specifies which networks can access the server.
SET BACKLOG	Specifies the server connection queue limits.
SET CONNECTED	Specifies the connection-request-received routine.
SET DISABLED-NODES	Specifies which OpenVMScluster nodes cannot execute the service.
SET ENABLED-NODES	Specifies which OpenVMScluster nodes can execute the service.
SET FLAGS	Specifies the flag bit mask for service operation control.
SET INIT	Specifies the initialize-service routine.
SET KEEPALIVE-TIMERS	Sets keepalive timers for a service.
SET LISTEN	Specifies the listen-for-connections routine.
SET LOG-ACCEPTS	Enables/disables successful connections logging.
SET LOG-FILE	Specifies the log message destination.
SET LOG-REJECTS	Enables/disables failed connections logging.
SET MAX-SERVERS	Specifies the service process limit.
SET PARAMETERS	Specifies service-dependent parameters.
SET PRIORITY	Specifies a OpenVMS priority for the created processes.
SET PROCESS	Specifies that the service is to run in an auxiliary master server process rather than in the main master server process.
SET PROGRAM	Specifies a OpenVMS file name for run or merged images.
SET REJECT-BY-DEFAULT	Enables/disables conditional connection rejection.
SET RECEIVE-BUFFER-SPACE	Specifies the size of the receive socket buffers.
SET REJECT-HOSTS	Specifies which hosts are not allowed service access.
SET REJECT-MESSAGE	Specifies a rejected connection message.
SET REJECT-NETS	Specifies which networks are not allowed service access.
SET SEND-BUFFER-SPACE	Specifies the size of the send socket buffers.
SET SERVICE	Specifies the perform-service routine.
SET SERVICE-NAME	Changes the service name.
SET SERVICE-TYPE	Sets the service type advertised for a particular service.
SET SOCKET-FAMILY	Specifies the service family address.
SET SOCKET-OPTIONS	Specifies the **setsockopt()** options.
SET SOCKET-PORT	Specifies the port for connection listening.
SET SOCKET-TYPE	Specifies the socket type.
SET USERNAME	Specifies the user name under which the selected service is started.
SET WORKING-SET-EXTENT	Specifies how much memory the process will be allowed to use if there are free pages available.
SET WORKING-SET-QUOTA	Specifies the maximum amount of memory the process can lock into its working set.

SERVER-CONFIG Command	Description
SHOW	Shows the current server configuration.
SHUTDOWN	Stops the master server process.
SPAWN	Invokes the DCL command or creates a subprocess.
STATUS	Shows the SERVER-CONFIG service status.
USE	Reads a server configuration file; same as **GET**.
VERSION	Shows the SERVER-CONFIG version.
WRITE	Writes the current server configuration; same as **SAVE**.

ADD

ADD — Adds a new service to the current server configuration and prompts you for an initial set of parameters for the service.

Format

`ADD` *`service`*

Parameter

`service`

Specifies the name of the service to add to the configuration.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ADD NNTP
[Adding new configuration entry for service "NNTP"]
Protocol: [TCP] TCP
TCP Port number: 119
Program to run: USER$DISK:[NNTP]NNTP_SERVER.EXE
[Added service NNTP to configuration]
[Selected service is now NNTP]
SERVER-CONFIG>
```

ATTACH

ATTACH — Detaches the terminal from the calling process and reattaches it to another process. Use the **SPAWN SHOW PROCESS /SUBPROCESSES** command to list the name of subprocesses. Use the DCL **LOGOUT** command to return to the original process. If the `IP$DISABLE_SPAWN` logical is enabled, **ATTACH** does not work.

Format

`ATTACH` *`process-name`*

Parameter

process-name

Specifies the name of a process to which you want your terminal attached. (Not all subprocesses can be attached; some testing may be required.)

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SPAWN
$ MM
MM>SPAWN SHOW PROCESS /SUB
...
There are 3 processes in this job:
_TWA42:
 PROC_1
 PROC_2 (*)
MM>ATTACH _TWA42:
SERVER-CONFIG>ATTACH PROC_1
MM>QUIT
$ LOGOUT
SERVER-CONFIG>
```

This example shows the use and exit of attached subprocesses.

1. The first command uses SPAWN to create a subprocess. MM is invoked from the DCL command line. Next, the **SPAWN SHOW PROCESS /SUB** command is used to list all the subprocess names. The display shows that three subprocesses are active. (Process `_TWA42:` is SERVER-CONFIG, `PROC_1` is MM, and `PROC_2` is the **SPAWN SHOW PROCESS /SUB** command.)
2. In the next command, the MM **ATTACH** command returns control to the SERVER-CONFIG process. From this utility, **ATTACH** returns control to MM. To exit, **QUIT** is invoked from MM, and **LOGOUT** is entered at the original spawned DCL command line; finally, control returns to SERVER-CONFIG. (If **SPAWN SHOW PROCESS /SUB** had been entered, only this command and the configuration processes would be active.)

COPY

COPY — Copies a given service entry. When a service is copied, the copy is disabled automatically. Enable the copy after changing any conflicting parameters, such as the port number.

Format

COPY *input-service output-service*

Parameters

input-service

Specifies the name of the service to duplicate.

`output-service`

Specifies the name of the service to create.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>COPY FINGER LOCALFINGER
[Adding service LOCALFINGER to configuration]
[Disabling service LOCALFINGER]
SERVER-CONFIG>
```

DELETE

DELETE — Deletes a given service from the current configuration. Once a service is deleted, all information about that service is removed. See the **DISABLE** command for disabling a service.

Format

`DELETE` *`service`*

Parameter

`service`

Specifies the name of the service to delete from the configuration.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>DELETE NNTP
SERVER-CONFIG>
```

DISABLE

DISABLE — Disables (removes) a given service from the current configuration. Unlike the **DELETE** command, all information about the service is retained and the service can be re-enabled at any time after a **DISABLE**.

Format

`DISABLE` *`service`*

Parameter

`service`

Specifies the name of the service to disable from the configuration.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>DISABLE NNTP
SERVER-CONFIG>
```

ENABLE

ENABLE — Enables a previously disabled service. (See **DISABLE** for information on disabling a service.)

Format

`ENABLE` *`service`*

Parameter

`service`

Specifies the name of the service to enable in the configuration.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ENABLE NNTP
SERVER-CONFIG>
```

EXIT

EXIT — Saves the current configuration, if it has been modified, then quits.

Format

`EXIT`

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ENABLE NNTP
SERVER-CONFIG>EXIT
[Writing configuration to IP$COMMON_ROOT:[IP] SERVICES.MASTER_SERVER]
$
```

GET

GET — Reads in a VSI TCP/IP server configuration file. After a **GET**, you can use the various configuration commands to modify this server configuration. (Functionally equivalent to **USE**.)

Format

GET *config_file*

Parameter

config_file

Specifies the name of the server configuration file to read in.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>GET ST_TMP:FOO.CONFIGURATION
[Reading in configuration from ST_ROO:[TMP]FOO.CONFIGURATION.1]
SERVER-CONFIG>
```

HELP

HELP — Invokes command help.

Format

HELP [*topics*]

Parameter

[*topics*]

Contains a space-delimited list of topics that begins with a topic followed by subtopics. The default topic is **HELP**.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>HELP ?
ADD       ATTACH   COPY     DELETE       DISABLE   ENABLE
EXIT      GET      HELP     NETCONTROL   PUSH      QUIT
RESTART   SAVE     SELECT   SET          SHOW      SHUTDOWN
SPAWN     STATUS   USE      VERSION      WRITE
SERVER-CONFIG>
```

NETCONTROL

NETCONTROL — Transfers control to a configuration manager subsystem that contacts the **NETCONTROL** server at local or remote sites. After invoking **NETCONTROL**, you can issue commands to the **NETCONTROL** server to affect IP$SERVER operations at that site.

Format

`NETCONTROL` [*`host`*]

Restriction

The **NETCONTROL** server is usually protected from unauthorized access by a restriction list.

Parameter

[*`host`*]

Specifies the name of the host to which to connect. If not specified, the default is the local host.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>NETCONTROL
Connected to NETCONROL server on "127.0.0.1"
<FLOWERS.COM Network Control 10.5 (nnn) at Mon 15-Mar-2017 7:42am-EST
NETCONTROL>
```

PUSH

PUSH — Starts and attaches a DCL subprocess. If a parent process exists, attach to it. To return from DCL, use the **ATTACH** or the **LOGOUT** command. To switch back from a DCL subprocess, use the **ATTACH** command. If the `IP$DISABLE_SPAWN` logical is set, **PUSH** does not work.

Format

`PUSH`

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>PUSH
$ LOGOUT
Process foobar_1 logged out at 16-Apr-2017 16:36:22.13
SERVER-CONFIG>
```

QUIT

QUIT — If the configuration file has been edited, **QUIT** prompts you to save the file before quitting.

Format

`QUIT`

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>QUIT
$
```

RESTART

RESTART — Kills the old master server (IP$SERVER) process and starts a new one. Any connections in progress are not interrupted. If the configuration has been modified since the last save, **RESTART** prompts you to save the configuration before restarting.

Format

```
RESTART
```

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>RESTART
%RUN-S-PROC_ID, identification of created process is 2060005c
SERVER-CONFIG>
```

SAVE

SAVE — Writes the current VSI TCP/IP server configuration to a server configuration file. (Functionally equivalent to **WRITE**.)

Format

SAVE *config_file*

Parameter

config_file

Specifies the name of the file to which to write the current VSI TCP/IP server configuration (by default, the same file from which the configuration was read).

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SAVE
[Writing configuration to IP$COMMON_ROOT:[IP]SERVICES.MASTER_SERVER.1103]
```

```
SERVER-CONFIG>
```

SELECT

SELECT — Selects which service will be modified by any subsequent **SET** commands.

Format

`SELECT` *`service`*

Parameter

`service`

Specifies the name of the service to select for modification.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT NNTP
[The Selected SERVER entry is not NNTP]
SERVER-CONFIG>
```

SET ACCEPT-HOSTS

SET ACCEPT-HOSTS — Specifies and maintains a list of hosts allowed access to the service. For IPv6 services IPv6 addresses and IPv4 addresses can be used. IPv4 addresses that are specified for an IPv6 service are treated as V4 mapped addresses.

Format

```
SET ACCEPT-HOSTS
```

Example

This example shows how to delete host 192.0.0.1 from the accept-hosts list, and add host 192.0.0.4.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET ACCEPT-HOSTS
Delete address "IP-192.0.0.1" ? [NO] Y
[Address "IP-192.0.0.1" deleted from TELNET]
Delete address "IP-192.0.0.2" ? [NO]
Delete address "IP-192.0.0.3" ? [NO]
You can now add new addresses for TELNET. An empty line terminates.
Add Address: 192.0.0.4
Add Address:
```

```
SERVER-CONFIG>
```

SET ACCEPT-NETS

SET ACCEPT-NETS — Invokes an interactive utility that prompts you for the addresses of networks that are allowed access to the selected service. Specify each network as follows: `IP_address [subnetmask]` When done, press **RETURN** at the "Add:" prompt. For more information about restricting access to services, see the *VSI TCP/IP Administrator's Guide: Volume II*.

Format

```
SET ACCEPT-NETS
```

Example

This example shows how to delete network address 192.0.0.0 from the accept-nets list, and add network address 128.1.0.0.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET ACCEPT-NETS
Delete address "IP-192.0.0.0" ? [NO] Y
[Address "IP-192.0.0.0" deleted from TELNET]
Delete address "IP-192.12.19.0" ? [NO]
You can now add new addresses for TELNET. An empty line terminates.
Add Address: 128.1.0.0
Add Address:
SERVER-CONFIG>
```

SET BACKLOG

SET BACKLOG — Specifies the number of server connections to queue up before refusing to accept additional connections when MAX-SERVERS is reached.

Format

SET BACKLOG *backlog*

Parameter

backlog

Specifies the number of connections to queue—but not process—while waiting for connections that are already running to exit.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
```

```
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET BACKLOG 5
[Backlog of TELNET set to 5]
SERVER-CONFIG>
```

SET CONNECTED

SET CONNECTED — Specifies the name of the internal IP$SERVER routine to call when a connection request is received.

Format

```
SET CONNECTED
```

[none]

[tcp_connected]

[udp_chargen]

[udp_connected]

[udp_connected_single]

[udp_daytime]

[udp_discard]

[udp_echo]

[udp_time]

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET CONNECTED UDP_CONNECTED_SINGLE
[Connected action of TELNET set to UDP_Connected_Single]
SERVER-CONFIG>
```

SET DISABLED-NODES

SET DISABLED-NODES — Specifies and maintains a list of OpenVMScluster nodes that cannot execute the service. The master server (IP$SERVER) can be tailored to enable or disable services on a per-node basis in a OpenVMScluster. Use the **SET ENABLED-NODES** or **SET DISABLED-NODES** commands to specify a list of OpenVMScluster nodes on which the service runs or does not run. The service must also be enabled via the **ENABLE** command. When entering the nodes to be disabled, use only the OpenVMScluster node name or the DECnet node name; do not use the IP address.

Format

```
SET DISABLED-NODES
```

Example

This example shows how to delete the node FLEET from the disabled-nodes list, and add the node DRAGO.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET DISABLED-NODES
Delete OpenVMScluster node "FLEET" ? [NO] Y
[Node "FLEET" deleted from TELNET]
Delete OpenVMScluster node "NINET9" ? [NO]
You can now add new OpenVMScluster nodes for TELNET. An empty line
 terminates.
Add OpenVMScluster node: DRAGO
Add Address:
SERVER-CONFIG>
```

SET ENABLED-NODES

SET ENABLED-NODES — Specifies and maintains a list of OpenVMScluster nodes that can execute the service. The master server (IP$SERVER) can be tailored to enable or disable services on a per-node basis in a OpenVMScluster. Use the **SET ENABLED-NODES** or **SET DISABLED-NODES** commands to specify a list of OpenVMScluster nodes on which the service runs or does not run. The service must also be enabled via the ENABLE command. When entering the nodes to be enabled, use only the OpenVMScluster node name or the DECnet node name; do not use the IP address.

Format

```
SET ENABLED-NODES
```

Example

This example shows how to delete the node DRAGO from the enabled-nodes list, and add the node FLEET.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET ENABLED-NODES
Delete OpenVMScluster node "DRAGO" ? [NO] Y
[Node "DRAGO" deleted from TELNET]
Delete OpenVMScluster node "NINET9" ? [NO]
You can now add new OpenVMScluster nodes for TELNET. An empty line
 terminates.
Add OpenVMScluster node: FLEET
Add Address:
```

```
SERVER-CONFIG>
```

SET FLAGS

SET FLAGS — Specifies a bit mask of flags that control the operation of the service. If you do not specify a flag for this command, all existing flags are cleared.

Format

SET FLAGS [*flag1* | *flag2* | *flag3*]

Flags

EUNICE_SERVER

Indicates the connection is not closed until the server process exits, instead of when the server process closes the connection. This flag is required for servers compiled under EUNICE, as the EUNICE runtime opens and closes the channel to the connection many times.

MANUAL

Indicates that the IP$SERVER process does not accept a CHAOSnet connection; instead, the created process performs the accept in a nonstandard way.

SNMP_MONITORED

Tells the service handling routines to connect to SNMP Agent X and provide information about the state of the service. You need to define `IP$SNMP_AGENTX 1` and add `AGENTX_PEER 127.0.0.1` in the `SNMPD.CONF` file. The values returned when SNMP_MONITORED flag is set are the Network Services Monitoring MIB (RFC2788). See the *VSI TCP/IP Administrator's Guide: Volume II* for an example of the Network Services Monitoring MIB. The following values are displayed within `enterprises.105.4.service_port`:

1. Service name
2. Service port number
3. Maximum servers allowed
4. Number of active servers
5. Number of times the service processing code has been called
6. Last I/O status value for the service (generally 0 or 1)

START_AUX_SERVER

Causes the main master server to start the auxiliary master server process automatically.

UCX_SERVER

Configures services written for VSI TCP/IP Services (formerly UCX) to work with VSI TCP/IP.

Examples

This example sets the EUNICE_SERVER flag.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET FLAGS EUNICE_SERVER
[TELNET flags set to <EUNICE_SERVER>]
SERVER-CONFIG>
```

This example clears all flags that are set on a service.

```
SERVER-CONFIG>SET FLAGS
[TELNET flags set to <NONE>]
SERVER-CONFIG>
```

SET INIT

SET INIT — Specifies the name of the internal IP$SERVER routine to call to initialize a service.

Format

```
SET INIT
```

[cluster_alias_init]

[merge_image]

[netcontrol_init]

[none]

[rpc_init]

[rpc_portmap_init]

[tcp_init]

[ucxqio_init]

[udp_init]

[viadecnet_init]

[viapsi_init]

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET INIT MERGE_IMAGE
[Init action of TELNET set to Merge_Image]
SERVER-CONFIG>
```

SET KEEPALIVE-TIMERS

SET KEEPALIVE-TIMERS — Sets keepalive timers for a service.

Format

```
SET KEEPALIVE-TIMERS idle-time prove-interval probe-count
```

Parameters

idle-time

Specifies the amount of time, in seconds, that a connection should be idle before the first keepalive probe is sent.

probe-interval

Specifies the number of seconds between keepalive probes.

probe-count

Specifies the number of probes that can be sent, with no reply from the other side of the connection, before the connection should be destroyed.

Description

Keepalives are useful in situations when other systems that connect to services provided by your system are subject to frequent crashing, resets, or power-offs (as with personal computers).

TCP/IP connections must pass through a three-way handshake sequence to be closed and removed from the connection table. However, if a connection is open but idle, and the remote system is shut off, reset, or crashes, the connection cannot be closed down until an attempt to communicate with the remote system is made. If an application or service does not do this, a keepalive probe can be used to ensure that these dormant connections are cleaned up.

If you set the `SO_KEEPALIVE` socket option for a service, but you do not explicitly set the KEEPALIVE-TIMERS, the default values are:

- *idle-time* 2 hours
- *probe-interval* 75 seconds
- *probe-count* 8

If you do not set the `SO_KEEPALIVE` socket option for a service, no keepalive probes will be sent for connections to that service.

SET LISTEN

SET LISTEN — Specifies the name of the internal IP$SERVER routine to call to listen for connections to the service.

Format

```
SET LISTEN
```

[none]

[tcp_listen]

[udp_listen]

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET LISTEN TCP_LISTEN
[Listen action of TELNET set to TCP_Listen]
SERVER-CONFIG>
```

SET LISTEN-ADDRESS

SET LISTEN-ADDRESS — Specifies the IP address that a service listens on. This parameter can be used to restrict a service to only accept incoming connections to a particular address configured on the system. Either an IPv4 or IPv6 address can be specified depending upon the socket-family that is set for the service.

Format

SET LISTEN-ADDRESS *192.168.1.1*

SET LISTEN-ADDRESS *2002:c0a8:101:1::1*

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET LISTEN-ADDRESS 192.168.1.1
[Server will listen on 192.168.1.1]
SERVER-CONFIG>
```

SET LOG-ACCEPTS

SET LOG-ACCEPTS — Specifies whether to log successful connections to the service.

Format

SET LOG-ACCEPTS *mode*

Parameter

mode

If mode is TRUE, accepted connections are logged to OPCOM or to the log file; if it is FALSE, accepted connections are not logged.

Example

```
$ IP CONFIGURE /SERVERS
```

```
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET LOG-ACCEPTS TRUE
SERVER-CONFIG>
```

SET LOG-FILE

SET LOG-FILE — Specifies the destination of log messages: a OpenVMS file name, or OPCOM to direct messages to the OpenVMS OPCOM process.

Format

`SET LOG-FILE` [*opcom*] [*file_spec*]

Parameter

[*file_spec*]

Specifies the file specification to which to write the audit records; specify OPCOM to write the audit records to OPCOM.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET LOG-FILE OPCOM
SERVER-CONFIG>
```

SET LOG-REJECTS

SET LOG-REJECTS — Specifies whether to log rejected connections to the service. A connection can be rejected because of the values of the REJECT-HOSTS, REJECT-NETS, and REJECT-BY-DEFAULT parameters.

Format

`SET LOG-REJECTS` *mode*

Parameter

mode

If mode is TRUE, rejected connections are logged to OPCOM or to the log file; if it is FALSE, rejected connections are not logged.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
```

```
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET LOG-REJECTS TRUE
SERVER-CONFIG>
```

SET MAX-SERVERS

SET MAX-SERVERS — Specifies the maximum number of service processes to allow at any one time. If this limit is reached, additional connections up to BACKLOG are accepted but are not processed until one of the previous connections completes.

Format

SET MAX-SERVERS *number*

Parameter

number

Specifies the maximum number of server processes to create for this service.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET MAX-SERVERS 4
[Max_Servers of TELNET set to 4]
SERVER-CONFIG>
```

SET PARAMETERS

SET PARAMETERS — Specifies service-dependent parameters. These parameters are passed to the initialization routine of built-in services. (This is normally not used for user-written services. The parameters and their arguments differ on a per-service basis.)

Format

SET PARAMETERS

Example

This example enables debugging for the Domain Name Service (DNS).

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT DOMAINNAME
SERVER-CONFIG>SET PARAMETERS
Delete parameter "bootfile IP$:DOMAIN-NAME-SERVICE.CONFIGURATION"? [NO]
You can now add new parameters for DOMAINNAME. An empty line terminates.
```

```
Add Parameter: debug 3
Add Parameter:
[Service specific parameters for DOMAINNAME changed]
SERVER-CONFIG>
```

SET PRIORITY

SET PRIORITY — Specifies the OpenVMS process priority to assign to created processes.

Format

SET PRIORITY *priority*

Parameter

priority

Specifies the OpenVMS process priority to assign to created processes.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT SMTP
SERVER-CONFIG>SET PRIOITY 5
[Priority of SMTP set to 5]
SERVER-CONFIG>
```

SET PROCESS

SET PROCESS — Specifies that the service is to run in an auxiliary master server process rather than in the main master server process. The **SET PROCESS** command should be used in conjunction with the SET FLAGS start_aux_server command, unless the administrator wants to start the auxiliary server manually.

Format

SET PROCESS *process_name*

Parameter

process_name

Specifies the name of the process to run in auxiliary master process mode.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
```

```
SERVER-CONFIG>SET FLAG start_aux_server
SERVER-CONFIG>SET PROCESS MNSERVER_1
SERVER-CONFIG>SELECT RLOGIN
SERVER-CONFIG>SET FLAG start_aux_server
SERVER-CONFIG>SET PROCESS MNSERVER_1
SERVER-CONFIG>SELECT RSHELL
SERVER-CONFIG>SET FLAG start_aux_server
SERVER-CONFIG>SET PROCESS MNSERVER_1
SERVER-CONFIG>
```

SET PROGRAM

SET PROGRAM — Specifies the OpenVMS file name of the image to run or merge.

Format

SET PROGRAM *file_spec*

Parameter

file_spec

Specifies the name of the file containing the server image.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT SMTP
SERVER-CONFIG>SET PROGRAM IP$:SMTP_SERVER.EXE
[Program to run for SMTP set to IP$:SMTP_SERVER.EXE]
SERVER-CONFIG>
```

SET REJECT-BY-DEFAULT

SET REJECT-BY-DEFAULT — Specifies whether to reject a connection from a host that does not match any of the ACCEPT-HOSTS, ACCEPT-NETS, REJECT-HOSTS, and REJECT-NETS lists.

Format

SET REJECT-BY-DEFAULT *mode*

Parameter

mode

If mode is TRUE, the default is to reject connections; if it is FALSE, the default is to accept them.

Example

```
$ IP CONFIGURE /SERVERS
```

```
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT SMTP
SERVER-CONFIG>SET REJECT-BY-DEFAULT TRUE
SERVER-CONFIG>
```

SET RECEIVE-BUFFER-SPACE

SET RECEIVE-BUFFER-SPACE — Specifies the size of the receive socket buffers.

Format

SET RECEIVE-BUFFER-SPACE *size*

Parameter

size

Specifies the size of the receive socket buffers for the selected service.

SET REJECT-HOSTS

SET REJECT-HOSTS — Specifies and maintains a list of hosts that are not allowed to access the service. For IPv6 services IPv6 addresses and IPv4 addresses can be used. IPv4 addresses that are specified for an IPv6 service are treated as V4 mapped addresses.

Format

SET REJECT-HOSTS

Example

This example shows how to delete host 192.0.0.1 from the reject-hosts list, and add host 192.0.0.4.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET REJECT-HOSTS
Delete address "IP-192.0.0.1" ? [NO] Y
[Address "IP-192.0.0.1" deleted from TELNET]
Delete address "IP-192.0.0.2" ? [NO]
Delete address "IP-192.0.0.3" ? [NO]
You can now add new addresses for TELNET. An empty line terminates.
Add Address: 192.0.0.4
Add Address:
SERVER-CONFIG>
```

SET REJECT-MESSAGE

SET REJECT-MESSAGE — Specifies a text string to send down the network connection when a service is rejected.

Format

SET REJECT-MESSAGE *string*

Parameter

string

This parameter is written down the network connection before closing the connection when a request is rejected.

Restriction

This parameter is ignored on UDP services.

Example

This example sets the rejection message to "Service refused."

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET REJECT-MESSAGE Service refused
SERVER-CONFIG>
```

SET REJECT-NETS

SET REJECT-NETS — Specifies and maintains a list of networks or subnetworks that are not allowed to access the service.

Format

SET REJECT-NETS

Description

SET REJECT-NETS invokes an interactive utility that prompts you for the addresses of networks that are not allowed to access the selected service.

Specify each network as follows: `IP_address [subnetmask]`. When done, press **RETURN** at the "Add:" prompt.

For more information about restricting access to services, see the *VSI TCP/IP Administrator's Guide: Volume II*.

Example

This example deletes network address 192.0.0.0 from the reject-nets list, and adds network address 128.1.0.0.

```
$ IP CONFIGURE /SERVERS
```

```
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET REJECT-NETS
Delete address "IP-192.0.0.0" ? [NO] Y
[Address "IP-192.0.0.0" deleted from TELNET]
Delete address "IP-192.12.19.0" ? [NO]
You can now add new addresses for TELNET. An empty line terminates.
Add Address: 128.1.0.0
Add Address:
SERVER-CONFIG>
```

SET SEND-BUFFER-SPACE

SET SEND-BUFFER-SPACE — Specifies the size of the send socket buffers.

Format

`SET RECEIVE-BUFFER-SPACE` *`size`*

Parameters

`size`

Specifies the size of the send socket buffers for the selected service.

SET SERVICE

SET SERVICE — Specifies the name of the internal IP$SERVER routine to call to perform the service.

Format

`SET SERVICE`

[internal_ftp]

[internal_pop2]

[internal_pop3]

[internal_telnet]

[no_service]

[none]

[run_program]

[tcp_daytime]

[tcp_time]

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET SERVICE INTERNAL_TELNET
[Service action of TELNET set to Internal_Telnet]
SERVER-CONFIG>
```

SET SERVICE-NAME

SET SERVICE-NAME — Changes the name of the service.

Format

`SET SERVICE-NAME` *`name`*

Parameter

`name`

Specifies the new service name.

Example

This example shows how to rename a service to "FOO."

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET SERVICE-NAME FOO
[Name of selected service changed to FOO]
SERVER-CONFIG>
```

SET SERVICE-TYPE

SET SERVICE-TYPE — For SPX- or IPX-based servers, sets the service type advertised for the service. This setting is generally not changed by users.

Format

`SET SERVICE-TYPE` *`service-type`*

Parameter

`service-type`

Specifies the service type for SAP advertisements. This value is a decimal number in the range of 1 through 6535.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT servicename
SERVER-CONFIG>SET SERVICE-TYPE n
[servicename service type is n]
SERVER-CONFIG>
```

SET SOCKET-FAMILY

SET SOCKET-FAMILY — Specifies the address family of the service.

Format

SET SOCKET-FAMILY *family*

Parameter

family

Specifies the new protocol family for this service.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET SOCKET-FAMILY AF_INET
[TELNET socket family is AF_INET]
SERVER-CONFIG>
```

SET SOCKET-OPTIONS

SET SOCKET-OPTIONS — Specifies socket options to be set via **setsockopt()**. See the *VSI TCP/IP Programmer's Reference* for more information on socket options.

Format

SET SOCKET-OPTIONS *options*

Parameter

options

Specifies a list of socket options separated by a vertical bar (|).

Example

```
$ IP CONFIGURE /SERVERS
```

```
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET SOCKET-OPTIONS SO_DEBUG | SO_KEEPALIVE
[TELNET socket options set to <SO_DEBUG | SO_KEEPALIVE]
SERVER-CONFIG>
```

SET SOCKET-PORT

SET SOCKET-PORT — Specifies the port number on which to listen for connections.

Format

SET SOCKET-PORT *port*

Parameter

port

Specifies the name or number of the port on which to listen.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TELNET
SERVER-CONFIG>SET PORT 10
[TELNET socket port is now 10]
SERVER-CONFIG>
```

SET SOCKET-TYPE

SET SOCKET-TYPE — Specifies the type of socket; for example, SOCK_STREAM (TCP) or SOCK_DGRAM (UDP).

Format

SET SOCKET-TYPE *type*

Parameter

type

Specifies the socket type to listen on, usually SOCK_ DGRAM (UDP) or SOCK_STREAM (TCP).

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT TFTP
```

```
SERVER-CONFIG>SET SOCKET-TYPE SOCK_DGRAM
[Socket type of TFFTP set to SOCK_DGRAM]
SERVER-CONFIG>
```

SET USERNAME

SET USERNAME — Specifies the user name under which the selected service is started.

Format

```
SET USERNAME username
```

Parameter

username

Specifies the name of the user under which the selected service is started.

SET WORKING-SET-EXTENT

SET WORKING-SET-EXTENT — Specifies how much memory the process will be allowed to use if there are free pages available.

Format

```
SET WORKING-SET-EXTENT
```

Examples

1.
```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5(nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT FTP
SERVER-CONFIG>SET WORKING-SET 2000
[Working_Set of FTP set to 2000]
SERVER-CONFIG>
```

2.
```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5(nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT FTP
[The Selected SERVER entry is now FTP]
SERVER-CONFIG>SET WORKING-SET-QUOTA 4096
[Working Set Quota of FTP set to 4096]
SERVER-CONFIG>SET WORKING-SET-EXTENT 8192
[Working Set Extent of FTP set to 8192]
```

SET WORKING-SET-QUOTA

SET WORKING-SET-QUOTA — Specifies the maximum amount of memory the process can lock into its working set.

Format

```
SET WORKING-SET-QUOTA
```

Examples

1.
```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT FTP
SERVER-CONFIG>SET WORKING-SET 2000
[Working_Set of FTP set to 2000]
SERVER-CONFIG>
```

2.
```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5(nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT FTP
[The Selected SERVER entry is now FTP]
SERVER-CONFIG>SET WORKING-SET-QUOTA 4096
[Working Set Quota of FTP set to 4096]
SERVER-CONFIG>SET WORKING-SET-EXTENT 8192
[Working Set Extent of FTP set to 8192]
```

SHOW

SHOW — Shows the current server configuration.

Format

```
SHOW
```

Qualifier

```
/FULL
```

Provides a more detailed listing.

Example

This example displays detailed information about SMTP.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SHOW/FULL SMTP
Service "SMTP"
TCP socket (AF_INET,SOCK_STREAM), Port 25
Socket Options = SO_KEEPALIVE
INIT() = TCP_Init
LISTEN() = TCP_Listen
CONNECTED() = TCP_Connected
SERVICE() = Run_Program
Program = "IP$:SERVER_SMTP.EXE"
```

```
SERVER-CONFIG>
```

SHUTDOWN

SHUTDOWN — Stops the master server (IP$SERVER) process. After a **SHUTDOWN**, any subsequent network service requests are rejected by VSI TCP/IP until the **RESTART** command is executed.

Format

SHUTDOWN

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SHUTDOWN
SERVER-CONFIG>
```

SPAWN

SPAWN — Executes a single DCL command, or if entered without options, starts a subprocess with the same effect as **PUSH**. To return from DCL, use the **LOGOUT** command. If the `IP$DISABLE_SPAWN` logical is set, **SPAWN** does not work.

Format

SPAWN [*command*]

Parameter

[*command*]

Specifies a command to execute. If you omit *command*, a DCL command line subprocess is created.

Qualifiers

/INPUT=*file-spec*

Specifies an input file to the command you enter with **SPAWN**.

/LOGICAL_NAMES

/NOLOGICAL_NAMES

Specifies that logical names and logical name tables are not copied to the subprocess.

/SYMBOLS

/NOSYMBOLS

Specifies that global and local names are not passed to the subprocess.

/WAIT

/NOWAIT

Returns control without waiting for the command to complete. Do not use this qualifier with commands that have prompts or screen displays.

/OUTPUT=*file-spec*

Specifies a file that retains the output of the command invoked with **SPAWN**. This qualifier only works when a single command is entered without creating a DCL subprocess. In addition, this qualifier is positional; you must enter it immediately after **SPAWN** or other qualifiers.

Examples

This example displays terminal information, captures the output in a file, then displays the information with the **TYPE** command.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SPAWN/OUTPUT=FOO. SHOW TERM
SERVER-CONFIG>SPAWN TYPE FOO.
...
```

This example invokes a command procedure.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SPAWN @COMPROC
...
```

This example displays help information about the SERVER-CONFIG utility. Use the **LOGOUT** command to return control to SERVER-CONFIG.

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SPAWN
$ HELP IP CONFIGURE /SERVER
...
$ LOGOUT
SERVER-CONFIG>
```

STATUS

STATUS — Shows the status of the VSI TCP/IP server configuration program.

Format

STATUS

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5(nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>STATUS
This is the VSI TCP/IP for OpenVMS server configuration program Version
 10.5(nnn)
There are 65/8192 entries in the current server configuration.
There is NO selected SERVER entry.
The configuration IP$:SERVICES.MASTER_SERVER is not modified.
SERVER-CONFIG>
```

USE

USE — Reads in a VSI TCP/IP server configuration file. After a **USE**, you can use the various configuration commands to modify the server configuration. (Functionally equivalent to **GET**.)

Format

```
USE config-file
```

Parameter

config-file

Specifies the name of the server configuration file to read in.

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>USE ST_TMP:FOO.CONFIGURATION
[Reading in configuration from ST_ROOT:[TMP]FOO.CONFIGURATION.1]
SERVER-CONFIG>
```

VERSION

VERSION — Displays the VSI TCP/IP server configuration program version and release information.

Format

```
VERSION
```

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>VERSION
```

```
This is the VSI TCP/IP for OpenVMS Server configuration program Version
 10.5 (nnn)
SERVER-CONFIG>
```

WRITE

WRITE — Writes the current server configuration file. (Functionally equivalent to **SAVE**.)

Format

`WRITE` *`config-file`*

Parameter

`config-file`

Specifies the name of the file to which to write the current VSI TCP/IP server configuration (by default, the same file from which the configuration was read).

Example

```
$ IP CONFIGURE /SERVERS
VSI TCP/IP for OpenVMS Server Configuration Utility 10.5 (nnn)
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>WRITE
[Writing configuration to IP$COMMON_ROOT:[IP]SERVICES.MASTER_SERVER.1103]
SERVER-CONFIG
```

Appendix A. Trademark and Copyright Notifications

This appendix contains a complete listing of trademarks and copyright notification contained in this manual.

The material in this document is for informational purposes only and is subject to change without notice. It should not be construed as a commitment by VMS Software, inc. VMS Software, inc. assumes no responsibility for any errors that may appear in this document.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The following third-party software may be included with your product and will be subject to the software license agreement.

in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by John Robert LoVerso.

Kerberos. Copyright © 1989, DES.C and PCBC_ENCRYPT.C Copyright © 1985, 1986, 1987, 1988 by Massachusetts Institute of Technology. Export of this software from the United States of America is assumed to require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting. WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

DNSSIGNER (from BIND distribution) Portions Copyright (c) 1995-1998 by Trusted Information Systems, Inc.

Portions Copyright (c) 1998-1999 Network Associates, Inc.

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 TRUSTED INFORMATION SYSTEMS DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL TRUSTED INFORMATION SYSTEMS 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.

ERRWARN.C. Copyright © 1995 by RadioMail Corporation. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of RadioMail Corporation, the Internet Software Consortium nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY RADIOMAIL CORPORATION, THE INTERNET SOFTWARE CONSORTIUM AND CONTRIBUTORS ``AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL RADIOMAIL CORPORATION 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,

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, TORT (INCLUDING NEGLIGENCE) OR STRICT LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions Copyright © 1980, 1982, 1985, 1986, 1988, 1989, 1990, 1993 by The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright © 1993 by Hewlett-Packard Corporation.

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, and that the name of Hewlett-Packard Corporation not be used in advertising or publicity pertaining to distribution of the document or software without specific, written prior permission. THE SOFTWARE IS PROVIDED "AS IS" AND HEWLETT-PACKARD CORP. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL HEWLETT-PACKARD CORPORATION 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.

Portions Copyright © 1995 by International Business Machines, Inc.

that file becomes a covered file. You may make a good-faith judgement as to where in this file the bootstrap license should appear.

4. The acronym "ISC", when used in this license or generally in the context of works covered by this license, is an abbreviation for the words "Internet Software Consortium."

5. A distribution, as referred to hereafter, is any file, collection of printed text, CD ROM, boxed set, or other collection, physical or electronic, which can be distributed as a single object and which contains one or more works covered by this license.

6. You may make distributions containing covered files and provide copies of such distributions to whomever you choose, with or without charge, as long as you obey the other terms of this license. Except as stated in (9), you may include as many or as few covered files as you choose in such distributions.

7. When making copies of covered works to distribute to others, you must not remove or alter the Bootstrap License. You may not place your own copyright message, license, or similar statements in the file prior to the original copyright message or anywhere within the Bootstrap License. Object files and executable files are exempt from the restrictions specified in this clause.

8. If the version of a covered source file as you received it, when compiled, would normally produce executable code that would print a copyright message followed by a message referring to an ISC web page or other ISC documentation, you may not modify the file in such a way that, when compiled, it no longer produces executable code to print such a message.

9. Any source file covered by this license will specify within the Bootstrap License the name of the ISC distribution from which it came, as well as a list of associated documentation files. The associated documentation for a binary file is the same as the associated documentation for the source file or files from which it was derived. Associated documentation files contain human-readable documentation which the ISC intends to accompany any distribution.

   If you produce a distribution, then for every covered file in that distribution, you must include all of the associated documentation files for that file. You need only include one copy of each such documentation file in such distributions.

   Absence of required documentation files from a distribution you receive or absence of the list of documentation files from a source file covered by this license does not excuse you from this requirement. If the distribution you receive does not contain these files, you must obtain them from the ISC and include them in any redistribution of any work covered by this license. For information on how to obtain required documentation not included with your distribution, see: http://www.isc.org/getting-documentation.html.

   If the list of documentation files was removed from your copy of a covered work, you must obtain such a list from the ISC. The web page at http://www.isc.org/getting-documentation.html contains pointers to lists of files for each ISC distribution covered by this license.

   It is permissible in a source or binary distribution containing covered works to include reformatted versions of the documentation files. It is also permissible to add to or modify the documentation files, as long as the formatting is similar in legibility, readability, font, and font size to other documentation in the derived product, as long as any sections labeled CONTRIBUTIONS in these files are unchanged except with respect to formatting, as long as the order in which the CONTRIBUTIONS section appears in these files is not changed, and as long as the manual page which describes how to contribute to the Internet Software Consortium (hereafter referred to as the Contributions Manual Page) is unchanged except with respect to formatting.

Documentation that has been translated into another natural language may be included in place of or in addition to the required documentation, so long as the CONTRIBUTIONS section and the Contributions Manual Page are either left in their original language or translated into the new language with such care and diligence as is required to preserve the original meaning.

10. You must include this license with any distribution that you make, in such a way that it is clearly associated with such covered works as are present in that distribution. In any electronic distribution, the license must be in a file called "ISC-LICENSE".

    If you make a distribution that contains works from more than one ISC distribution, you may either include a copy of the ISC-LICENSE file that accompanied each such ISC distribution in such a way that works covered by each license are all clearly grouped with that license, or you may include the single copy of the ISC-LICENSE that has the highest version number of all the ISC-LICENSE files included with such distributions, in which case all covered works will be covered by that single license file. The version number of a license appears at the top of the file containing the text of that license, or if in printed form, at the top of the first page of that license.

11. If the list of associated documentation is in a separated file, you must include that file with any distribution you make, in such a way that the relationship between that file and the files that refer to it is clear. It is not permissible to merge such files in the event that you make a distribution including files from more than one ISC distribution, unless all the Bootstrap Licenses refer to files for their lists of associated documentation, and those references all list the same filename.

12. If a distribution that includes covered works includes a mechanism for automatically installing covered works, following that installation process must not cause the person following that process to violate this license, knowingly or unknowingly. In the event that the producer of a distribution containing covered files accidentally or wilfully violates this clause, persons other than the producer of such a distribution shall not be held liable for such violations, but are not otherwise excused from any requirement of this license.

13. COVERED WORKS ARE PROVIDED "AS IS". ISC DISCLAIMS ALL WARRANTIES WITH REGARD TO COVERED WORKS INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

14. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OF COVERED WORKS.

Use of covered works under different terms is prohibited unless you have first obtained a license from ISC granting use pursuant to different terms. Such terms may be negotiated by contacting ISC as follows:

Internet Software Consortium

950 Charter Street

Redwood City, CA 94063

Tel: 1-888-868-1001 (toll free in U.S.)

Tel: 1-650-779-7091

Fax: 1-650-779-7055

Email: info@isc.org

Email: licensing@isc.org

DNSSAFE LICENSE TERMS

This BIND software includes the DNSsafe software from RSA Data Security, Inc., which is copyrighted software that can only be distributed under the terms of this license agreement.

The DNSsafe software cannot be used or distributed separately from the BIND software. You only have the right to use it or distribute it as a bundled, integrated product.

The DNSsafe software can ONLY be used to provide authentication for resource records in the Domain Name System, as specified in RFC 2065 and successors. You cannot modify the BIND software to use the DNSsafe software for other purposes, or to make its cryptographic functions available to end-users for other uses.

If you modify the DNSsafe software itself, you cannot modify its documented API, and you must grant RSA Data Security the right to use, modify, and distribute your modifications, including the right to use any patents or other intellectual property that your modifications depend upon.

You must not remove, alter, or destroy any of RSA's copyright notices or license information. When distributing the software to the Federal Government, it must be licensed to them as "commercial computer software" protected under 48 CFR 12.212 of the FAR, or 48 CFR 227.7202.1 of the DFARS.

You must not violate United States export control laws by distributing the DNSsafe software or information about it, when such distribution is prohibited by law.

THE DNSSAFE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER. RSA HAS NO OBLIGATION TO SUPPORT, CORRECT, UPDATE OR MAINTAIN THE RSA SOFTWARE. RSA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO ANY MATTER WHATSOEVER, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

If you desire to use DNSsafe in ways that these terms do not permit, please contact:

RSA Data Security, Inc.

100 Marine Parkway

Redwood City, California 94065, USA