TSO Diagnostic Commands

This chapter describes the diagnostic commands available in Cisco IOS for S/390. The chapter is organized into the following sections.

- Troubleshooting Client and Server FTP3

  Describes guidelines for problem isolation with the Cisco IOS for S/390 FTP compatibility.

- ACTEST

  A Customer Support command to debug the Cisco IOS for S/390 commutator.

- DNRGET

  A command to query the Domain Name Resolver (DNR).

- NETSTAT/SYSSTAT

  The diagnostic command to monitor and manage Cisco IOS for S/390 activity.

- OSPFMON

  A command to request information for routes known to OSPF.

- PING

  A command to determine if a host is active on the network.

- RIPQUERY

  A command to request all routes known by a RIP gateway.

- RPCINFO

  A command to query an RPC server to obtain information about program names and program numbers for a specified host.

- TCPEEP

  A command to invoke the Cisco IOS for S/390 packet trace program and diagnose remote host communication problems.

- TRACE

  Trace command enhancements to collect TCP/IP data and display it on a terminal or send it to an external writer.

- TRACERT

  A command to print the route that packets take to get to a network host.

Troubleshooting Client and Server FTP3

This section describes problem isolation in the Cisco IOS for S/390 FTP compatibility feature. Troubleshooting information is divided into server and client. The client is a three-party model; it therefore requires both server and client documentation.

General Guidelines for Problem Isolation

When encountering FTP3 problems, start with the following problem resolution process.

1. Determine the level of maintenance that is running.
2. Enable the DEBUG or TRACE option and rerun the failing command.
3. Run a network trace.
4. Look at the T01LOG job output.

Resolving Server FTP Problems

For incorrect input through FTP, collect the following:

- A description of the action preceding the failure and the condition of the system (such as 100 terabyte transfer).
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including
  - the JES log
  - any PSNAP data
  - T01LOG
  - any enabled traces
- A log or screen print of the failing transfer, including any messages.
- Any client software involved, including the name of the vendor, the release number, the direction of transfer, the file format; and the contents (if requested).
- Whether the failure is reproducible or random. If reproducible, obtain a TCPEEP TCP trace.

Resolving Client FTP Problems

For incorrect input through FTP, collect the following information:

- A description of the action preceding the failure and the condition of the system (such as 100 terabyte transfer).
- The job output of the Cisco IOS for S/390 gateway at the time of failure, including
  - the JES log
  - any PSNAP data
  - T01LOG
  - any enabled traces
- A log or screen print of the failing transfer, including any messages.
- Turn on the TRACE or DEBUG parameters. The recommended procedure is to run the command in batch mode and save the recorded trace output for problem resolution.

- Any client software involved, including the name of the vendor, the release number, the direction of transfer, the file format; and the contents (if requested).
- Whether the failure is reproducible or random. If reproducible, obtain an TCPEEP TCP trace.

ACTEST

The Telnet command **ACTEST** invokes the ACTEST parameter and debugging interface to the Cisco IOS for S/390 commutator. A generalized terminal I/O interface implements it. It is generally used only on advice of Customer Support.

To invoke **ACTEST**

- From Server Telnet, through an ATPUT/ATGET interface, select ACTEST at the Server Telnet command screen (port 23). It requests logon and calls PACCESS to verify privileges.
- From User Telnet (TSO), use VTAMTEST as the remote host name. The path requires a local logon. It is particularly useful when there is a problem with TCP/IP, since no network I/O is required.

This list describes the **ACTEST** options:

ABEND	Abruptly terminates the entire commutator.
ACCOM *length*	Lists length (in bytes) of the main control block.
ALIST *address length*	Displays storage at address within unprotected areas of the address space for length bytes; tries to print the information as ASCII; specifies address and length in hex.
CATV *pta_1 pta_2*	Copies the application transfer vector (PTAATRV) from the PTA at address pta_2 to the PTA at address pta_1.
COMMANDS	Lists all the commands known to ACTEST.
END	Terminates the ACTEST ptask normally.
EQUATE *symbol address length*	Defines a symbol to be used by LIST.
EXECUTE *slot*	Issues the commands saved in slot by the **REMEMBER** command.
FREEZE	Issues a **PWAIT FREEZE** and halts execution of the ACTEST ptask until external action is taken to unblock the ptask.
HELP *command*	Displays the syntax for command.
LIST *address length*	Displays storage at address within unprotected areas of the address space for length bytes. Tries to print the information as EBCDIC.
MODULES *mask*	Lists the loaded modules or ESDs that have names matching mask (* is a wild card character).

PATTACH *module parms*

Creates a new control fork to pass control to the indicated module and to pass it the indicated parameter string in the form defined for the EXEC PARM field; each fork is identified by a control block called the Pseudo Task Area (PTA).

PDELETE *name*

Removes from storage the module loaded by PLOAD called name.

PDETACH *pta*

Destroys the PTA at address pta that was created by PATTACH. Explicitly forces PTA at a given address to exit.

Use this to install a new load module within the running Cisco IOS for S/390 when the old module has been PLOADed by a permanent ptask. Issue PDETACH for the ptask to free the old module, and issue a new PATTACH to bring in the new module.

PLOAD *name options*

Loads the program module name into main storage where it stays until you exit from ACTEST. The possible options are: ESD, FETCH, FREE, HOLD, SYS, and USE.

PPOST *ecb code*

Marks event ecb complete with the code *code*.

PPOST *pta event_list*

Marks pta as no longer blocked on certain events. Possible events are ATTN, CLOSE, CORE, INPUT, OPEN, OUTPUT, and USER.

PSTATUS ANCESTOR *pta*

Lists all ancestors of the PTA at address pta.

PSTATUS DESCENDENT *pta*

Lists all descendents of the PTA at address pta.

PSTATUS PARENT *pta*

Lists all parents of the PTA at address pta.

PSTATUS PROCESS *mask*

Lists all the PTAs that have names matching the mask (* is a wild character).

PSTATUS SON *pta*

Lists all sons of the PTA at address pta.

PTALIST *mask*

Lists all the PTAs that have names matching the mask along with the status and contents of the registers of the PTA (* is a wild card character).

The command **PTA I** displays the status of all PTAs whose names begin with I (for example, INPTASK, IMPIN, IMPOUT, etc.). The PTA display formats the dispatchability and semaphore bits in the first word of the PTA and shows the registers at the time of the last P- service call (normally PWAIT).

PWAIT *integer*

Suspends the ACTEST ptask to let the commutator dispatch other ptasks (in other words, this lets Cisco IOS for S/390 execute for a time specified by integer). Here integer is in units of 0.01 seconds. To specify a decimal integer, follow the integer with a period.

REMEMBER *slot command_line*

Saves the command_line in slot for later execution. The text of command_line is saved and later given to the parse routines by the **EXECUTE** command. Saved commands can include nested **EXECUTE** commands. There are sixteen slots numbered 0 - 15.

RESOURCE *PTA_address mask*

Displays those resources of the PTA at the specified address that have names matching mask (* is a wild character in mask). If mask is not specified, all resources owned by the PTA display.

Examples

The following example lists the PCOREd areas belonging to the PTA at address 56A00.

```
RESOURCE 56A00 PCORE or RES 56A00 PC
```

The following example displays the AEXIT resource for the specified PTA. This includes all the TCP/IP and environmental control blocks.

```
RES 56A00 AEXIT or RES 56A00 AE
```

The following example displays all the resources owned by the specified PTA.

```
RES 56A00
```

RUN *transfer_initialization_parameters*

Starts an application program. The module identified by transfer is loaded into core and treated as the Application Transfer Vector (ATRV). The module identified by initialization is then forked by PATTACH (specifying the option that the new ptask is to have no parents) with the ATRV and given the parameters in EXEC PARM form.

SHOWREM *slot*

Displays the command line saved in slot by the **REMEMBER** command.

SPOOLDMP ON | OFF

Specifies whether ptask SPOOL#4 should take a dump at certain task termination events (open, close, dynamic allocation, etc.). SPOOLDMP entered without any parameters will display the current setting.

SRUN *pta*

Makes the PTA at address pta ready to run.

START *pta*

Revives the PTA at address pta that was suspended with **STOP**.

STOP *pta*

Suspends the PTA at address pta from running.

STORE *data*

Replaces storage with data at the last address displayed with the **LIST** command.

SUBPOOLS

Display allocated, used, and free virtual storage in all used subpools above and below the 16MB “line”, including totals.

SVC99

Activates the extended error processing from SVC99. These extended messages will be issued to the master console by the SVC99 error processing facility.

SYSDUMP ON | OFF

Specifies whether a system dump should be produced for commutator ABEND conditions, in addition to the formatted SNAP dump that is normally produced. The system dump goes to the SYSUDUMP, SYSABEND, or SYSMDUMP data set.

Default: OFF

TASKS *mask*

Alternate for the **PSTATUS PROCESS** mask command. Gives a one line status display for each ptask whose name begins with the string mask. If mask is omitted, the status display lists all current ptasks.

VLT ON | OFF

VLT tracing. VLT OFF stops VLT tracing; VLT ON starts VLT tracing. Use module AMDUSR to format the GTF records written by the VLT trace.

Default: VLT OFF

WHERE *address*

Locates the given address within the current set of loaded modules, if possible, and prints its relative position.

XSEC *options* ON | OFF

Activates or deactivates monitoring of selected security events. Those options can also be specified on the SECURITY statement in member IJTCFGxx. The global system security options can be turned on or off by specifying ON or OFF.

- ACEE - Monitors ACEE activity.
- ACSECP -Monitors entry to the security call module that performs all security checking within Cisco IOS for S/390.
- COMMAND - Monitors command authorization calls. The COMMAND option aids in the debugging of COMMAND security problems within Cisco IOS for S/390.
- DATASET - Monitors data set authorization calls. The DATASET option aids in the debugging of DATASET security problems within Cisco IOS for S/390.
- LOGOFF - Monitors departure from the system. The LOGOFF option aids in the debugging of sign-off security problems within Cisco IOS for S/390.
- LOGON - Monitors attempts to gain entry to the system. The LOGON option aids in the debugging of sign-on security problems within Cisco IOS for S/390.

DNRGET

The **DNRGET** TSO diagnostic command invokes the command processor that allows direct queries to be made to the Domain Name Resolver (DNR). DNRGET can issue all sixteen combinations of domain name requests supported by the DNR. Also, the full range of options that the DNR can support can be specified.

Each reply returned by the DNR is formatted and displayed. The format of the returned data is dependent on the type and mode of the request. Any conditional completion codes or error codes are formatted and displayed. Along with these codes, a textual message explaining the code is displayed.

DNRGET executes as a TSO command processor. All messages that are written and done with TPUTs restricting its use to interactive TSO users only. Also, the TSO profile option PROMPT must be set if **DNRGET** is to prompt for correct operands in the event of an error.

DNRGET can not be invoked with the **TSO CALL** or **TSOEXEC** commands. Therefore the LINK data set must either be in your system's link list or in the STEPLIB DD concatenation of your TSO JCL procedure.

If, after invocation, **DNRGET** waits too long for a reply from the DNR, press **PA1** to interrupt its execution without causing any errors.

The syntax of the **DNRGET** command is as follows:

DNRGET *request_type request_mode*

request **[Block | Noblock] [Copy | Nocopy] [Local | Global] [SIze(***count***)]**

[SYsid(*subsystem_id***)] [Time(***seconds***)]**

Note At a minimum, you must specify the letters that are capitalized of the request parameters.

Syntax Description

request_type Specifies the type of Domain Name System information that is being requested. These are the valid values:

- HOST—Resolves host names and Internet addresses.
- HOSTINFO—Returns the CPU type and operating system of a host.
- HOSTSERV—Returns which services a host provides.
- NETWORK—Resolves network names and Internet network numbers.
- NS—Returns list of authoritative name servers for domain.
- PROTOCOL—Resolves transport protocol names and Internet protocol numbers.
- ROUTE—Returns mail routing information for a host.
- RPC—Resolves RPC program names and numbers.
- SERVICE—Resolves well known protocol services.
- SOA—Returns start of authority data for a domain.

request_mode Specifies the format of the request data. These are the valid values:

- BYALIAS—The request field is a host name ALIAS. This option returns a replacement name for the requested host name.
- BYNAME—The request field is either a host name, network name, protocol name/service name or protocol name. BYNAME is valid with any request type.
- BYVALUE—The request field is either an Internet number in nnn.nnn.nnn.nnn format, a TCP/UDP service number, or a protocol number. BYVALUE is only valid with request types HOST, NETWORK, SERVICE, and PROTOCOL.

request

Specifies the data that the Domain Name Resolver used as input. Its format is dependent on the request_type and request_mode arguments. These are the valid values:

- Block | Noblock—Block specifies that a response is not returned until either the request has completed or the time or retry limits have been exceeded. Noblock specifies that a response is returned immediately. If the data to complete the request is not in the cache, an error is returned immediately, but the DNR continues to try to resolve the request in order to update the cache.

  This argument only applies if Global is specified.

  Default: Block

- Copy | Nocopy—Copy specifies that if data to resolve a request is in the DNR cache it uses it to satisfy the request. Nocopy specifies that DNR ignores the data in the cache and queries domain name servers for data to resolve the request.

  This parameter only applies if Global is specified.

  Default: Copy

- Local | Global—Local specifies that the DNR use only configuration data to resolve requests. Global specifies that the DNR query Domain Name Servers for the necessary data to resolve the request.

  Default: Global

- SIze(*count*)—Specifies the number of entries to be returned by the DNR when the result field has more than one item in the reply. count is the maximum number of entries to be returned. 0 (zero) specifies that the maximum number of entries is determined by the return buffer size, which cannot be dynamically changed.

  Default: 0 (zero)

- SYsid (*subsysid*)—Specifies the subsystem ID of the Cisco IOS for S/390 job or started task in which the DNR is active. subsystem_id is the four character MVS subsystem ID.

  Default: ACSS

- Time(*seconds*)—Specifies the maximum amount of time the DNR is allowed to resolve a request. seconds is the maximum amount to process the request in seconds. 0 (zero) specifies that there is no time limit to resolve the request.

  Default: 30 seconds

NETSTAT/SYSSTAT

The **NETSTAT** and **SYSSTAT** diagnostic commands invoke the user information and operator control interface for Cisco IOS for S/390. The diagnostic commands are used to monitor and manage Cisco IOS for S/390 activity.

The TCP task group must be active. System security adjustments should be made if the new management commands require authorization

Information retrieval is from a running, operational gateway, so commands associated with the NETSTAT facility can originate outside the gateway address space.

Commands can originate from the following.

- The system console operator
- A TSO user
- A remote user via Telnet

NETSTAT commands that interrogate only are unrestricted commands; management commands that result in dynamic modification of internal control blocks or session behavior require authorization, unless the source is the system console operator.

Note SYSSTAT commands entered under TSO are secure based on the TSO userid.

Command Syntax

The external syntax of commands varies slightly, depending on the origin. Operator commands start with some sequence that identifies the target address space.

For example, to request a NETSTAT connection display, use one of the following operator commands:

F *jobname***,NETSTAT CONN**

or

%NETSTAT CONN

where **%** is the subsystem recognition character.

A remote Telnet user would use the following command:

NETSTAT CONN

A TSO version may include TSO-specific keywords:, as shown by the following

NETSTAT CONN SSID(ABCD)

Dotted Notation IP Address Wildcarding (*)

In most cases where a dotted notation IP address is accepted, wildcarding or omission adheres to the following syntax:

nnn <.*nnn* | * > <.*nnn* | * > <.*nnn* | * >

The usual dotted notation consists of four numeric components, each between 0-255, and separated by a period. For example:

111.112.113.4

If a component is omitted, right to left, it is replaced with 0, so 111.112.113 will be interpreted as a specific IP address of 111.112.113.000, and 111.112 will be interpreted as 111.112.000.000.

If a component is replaced with an asterisk (*), it will be interpreted as a non-specific wildcard. So 111.112.113.* will satisfy any IP address match starting with 111.112.113, and 111.112.* will satisfy any IP address starting with 111.112.

Wildcarding must be applied to an entire component. In other words, 11* is not valid. Both wildcarding and omission must be rightmost in the IP address string. That is, 111.*.113.4 is not valid.

TSO Command Common Parameters

The DDNAME(*name*) and SSID(*name*) parameters are common to all TSO commands.

NETSTAT CONN DDNAME(*ddname* **) SSID(** *subsysname* **)**

DDNAME(*ddname*)

If the standard TSO output definition SYSTSPRT is not appropriate, an output file can be pre-allocated with an LRECL size of 120 bytes. Its DD name can then be used as a parameter to the TSO **NETSTAT** or **SYSSTAT** command. This option is particularly useful if archiving is required, and browsing a disk file may be convenient.

If output is directed to a file specified by the DDNAME(*name*) parameter, all message lines are date and time stamped.

SSID(*subsysname*)

Specifies the subsystem ID.

NETSTAT Diagnostic Commands

The NETSTAT diagnostic commands are described following.

ARP

Display the status of devices that have responded to a request or are pending a request. The request status is retrieved from cache.

NETSTAT ARP[IPNAME] [MEDIA(*name* **)] [MSGID | NOMSGID]**

Syntax Description

null Display all devices.

IPNAME Attempt to resolve the symbolic IP host name. If not resolvable, dotted notation displays.

MEDIA(*name* **)** Limit the display to a specific media, as defined in TCPCFGxx.

MSGID | NOMSGID Controls the display of the 9-character message prefix.

- For Telnet user, NOMSGID is the default; use MSGID to turn on the message display.
- For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display.
- For operator requests, the prefix is always displayed; this parameter has no effect.

The following messages result from the **NETSTAT ARP** command:

T01NT024I T01NT025I T01NT026I

The status is defined as follows:

- Valid – a response was received from a device request
- Dead – the requested device did not respond
- Static – the device is defined in the TCPCFGxx file
- Router – the requested device is a router
- Pending resolution – a request has been made but not response was received

HELP

Displays a summary of the **NETSTAT** and **SYSSTAT** commands.

NETSTAT HELP [*command***] [MSGID | NOMSGID]**

Syntax Description

null	Display a list of all the NETSTAT commands.
command	Display a specific command, description and syntax.
MSGID | NOMSGID	Controls the display of the 9-character message prefix. • For Telnet user, NOMSGID is the default; use MSGID to turn on the message display. • For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display. • For operator requests, the prefix is always displayed; this parameter has no effect.

The following messages result from the **NETSTAT HELP** command:

T01NT021I T01NT022I

CNFG

Display configuration information in one of the following formats.

NETSTAT CNFG [MSGID]

NETSTAT CNFG LNI [CETI | CLAW] [MSGID]

NETSTAT CNFG TCP | UDP | RAW | ALL [MSGID]

Syntax Description

null Display the basic network in an outline form.

MSGID Controls the display of the 9-character message prefix.

- For Telnet user, the default is no message. Use MSGID to turn on the message display.
- For TSO users, MSGID is the default; using no message omits the prefix from the display.
- For operator requests, the prefix is always displayed; this parameter has no effect.

LNI [CETI | CLAW] Display configuration information from the TCPCFG*xx* statement. The LNI selection displays Local Network Interface information, and TPI displays Transport Provider Information variables. If the CETI option is selected for a CETI device, then 4 additional messages are displayed. If the CLAW option is selected for a CLAW device, then 3 additional messages are displayed.

TCP | UDP | RAW Protocol identifiers, TCP, UDP, and RAW can be used to limit the display to specific protocols, otherwise all protocols are displayed.

The following messages result from the **NETSTAT CNFG** command:

T01NT028I
T01NT029I
T01NT030I
T01NT031I
T01NT032I
T01NT033I
T01NT034I
T01NT035I
T01NT036I
T01NT037I
T01NT038I
T01NT039I
T01NT040I
T01NT041I
T01NT042I
T01NT057I
T01NT058I
T01NT059I
T01NT060I
T01NT061I
T01NT062I
T01NT064I
T01NT065I
T01NT066I
T01NT067I
T01NT086I
T01NT087I
T01NT088I
T01NT089I

CONN

Gives connection summary information for specified connection

NETSTAT CONN

[COUNT] [FULL] [MSGID | NOMSGID] [SEP] [TCP | UDP | RAW]

[LOCPORT(*nnn)***] [REMPORT(***nnn)***]**

[LOCHOST(*ip_addr)***] [REMHOST(***ip_addr*)**]**

[JOB(*name)***] [PLU(***name*)**] [SLU(***name)***] [SESSION(***nnn*)**]**

Syntax Description

COUNT Specifies that a one-line connection count be displayed

FULL Specifies that the full output be displayed, with no filtering. This can be approximately 23 lines per connection.

MSGID | NOMSGID Controls the display of the 9-character message prefix.

- For Telnet user, NOMSGID is the default; use MSGID to turn on the message display.
- For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display.
- For operator requests, the prefix is always displayed; this parameter has no effect.

SEP Specifies that a separator, a single line with an asterisk (*) be placed between connections. This is useful when many connections are being displayed.

TCP | UDP | RAW Protocol identifiers. TCP, UDP, and RAW can be used to limit the display to specific protocols, otherwise all protocols are displayed.

LOCPORT(*nnn*) Display one or more sessions, using a local port number (*nnn*).

REMPORT(*nnn*) Display one or more sessions, using a remote port number (*nnn*).

LOCHOST(*ip_addr*) Display a connection to a local host. *ipaddr* is the IP address of the host. The IP address must be entered in dotted notation.

REMHOST(*ip_addr*) Display a connection to a remote host. *ipaddr* is the IP address of the host. The IP address must be entered in dotted notation.

JOB(*name*) Specifies that the output be filtered for the jobname given.

PLU(*name*) Specifies that the output be filtered for the Primary LU given.

SLU(*name*) Specifies that the output be filtered for the Secondary LU given.

SESSION(*nnn*) Display a session. *nnn* represents a specific session number.

The following messages result from the **NETSTAT CONN** command:

T01NT063I T01NT054I T01NT055I T01NT056I

ROUTE

Displays the routing table entries.

NETSTAT ROUTE [IPNAME] [MSGID | NOMSGID]

Syntax Description

null	Display a list of all the NETSTAT commands.
IPNAME	Attempt to resolve the symbolic IP host name. If not resolvable, dotted notation will display.
MSGID | NOMSGID	Controls the display of the 9-character message prefix. • For Telnet user, NOMSGID is the default; use MSGID to turn on the message display. • For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display. • For operator requests, the prefix is always displayed; this parameter has no effect.

The following messages result from the **NETSTAT ROUTE** command:

T01NT044I T01NT045I

USER

Displays information about connected users.

NETSTAT USER [SESSION(*nnn*)] [IPNAME] [MSGID | NOMSGID]

or

NETSTAT USER [REMHOST(*ip_addr*)] [LOCPORT(*nnn*)] [IPNAME]
[MSGID | NOMSGID]

Syntax Description

null
List all users

SESSION(*nnn*)
Limit the display to a specified session.

REMHOST(*ip_addr*)
Limit the display to the entries associated with a specific host. The *ipaddr* can be specified as either dotted notation, or a symbolic string, and an ending asterisk (* can be used as a wildcard indicator. The **REMHOST(***ip_addr***)** and **LOCPORT(** *nnn* **)** options can be used together or separate.

LOCPORT(*nnn*)
Limit the display to the entries associated with a specific port.

IPNAME
Attempts to resolve and display actual hostname instead of IP address.

MSGID | NOMSGID
Controls the display of the 9-character message prefix.

- For Telnet user, NOMSGID is the default; use MSGID to turn on the message display.
- For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display.
- For operator requests, the prefix is always displayed; this parameter has no effect.

The following messages result from the **NETSTAT USER** command:

T01NT046I T01NT047I

SYSSTAT Diagnostic Commands

The SYSSTAT diagnostic commands include all of the NETSTAT commands, plus the commands **CANC** and **CHANGE**. These are described following.

CANC

Cancel a specified session.

SYSSTAT CANC [SESSION(*nnn***)] [MSGID | NOMSGID]**

or

SYSSTAT CANC [TCP | UDP | RAW] REMHOST(*ip_addr***) | LOCPORT(***nnn***)**
[MSGID | NOMSGID]

Syntax Description

SESSION(*nnn*)	Cancel a specific session number (*nnn).*
TCP | UDP | RAW	If cancellation is not by specific session number, one of these protocol identifiers must be specified.
LOCPORT(*nnn*)	Cancel one or more sessions, using a local port number. *nnn* represents a specific port number.
REMHOST(*ip_addr*)	Cancel a connection to a remote host. *ipaddr* is the IP address of the host. The IP address must be entered in dotted notation.
MSGID | NOMSGID	Controls the display of the 9-character message prefix. • For Telnet user, NOMSGID is the default; use MSGID to turn on the message display. • For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display. • For operator requests, the prefix is always displayed; this parameter has no effect.

Caution Cancellation by a specific session number will only cancel a single session. The other option, cancellation by protocol/IP/port, can terminate multiple sessions.

CHANGE

Change TCP variables.

SYSSTAT CHANGE [CWIND(*nnn***) SESSION(***nnn***)] [THRS(***nnn***) SESSION(***nnn)***]**
[RTT(*nnn***) SESSION(***nnn***)] [DEBUG(ON | OFF)] [SESSION(***nnn***)]**
[MSGID | NOMSGID]

Syntax Description

CWIND(*nnn***) SESSION(***nnn***)**

Change the congestion window size for a specified session. Both variables are required.

THRS(*nnn* **) SESSION(** *nnn* **)**

Change the slow start threshold for a specified session. Both variables are required.

RTT(*nnn* **) SESSION(** *nnn* **)**

Change the round trip start time for a specified session. Both variables are required.

DEBUG(ON | OFF)

Used in TCP connections to produce debugging messages to the T01LOG data set. In order to use this feature, the TCP task group must have the TEST flag set on. An operator command, or startup script, may set this flag, using the command **/SET TEST ON TGB(TCP)**. Similarly the flag may be set off using the command **/SET TEST OFF TGB(TCP)**. Once this flag is on, the individual DEBUG flags of each connection are queried.

Using this method to diagnose TCP problems is not recommended, as large volumes of messages may be produced. These messages are directed to the T01LOG data set. The messages produced are in the range of T01TC9*xx*.

MSGID | NOMSGID

Controls the display of the 9-character message prefix.

- For Telnet user, NOMSGID is the default; use MSGID to turn on the message display.
- For TSO users, MSGID is the default; use NOMSGID to omit the prefix from the display.
- For operator requests, the prefix is always displayed; this parameter has no effect.

OSPFMON

The **OSPFMON** (ospf_monitor) TSO diagnostic command can request information for routes known to OSPF.

Note **OSPFMON** executes as a TSO command processor. All messages that are written are done with TPUTs, restricting its use to interactive TSO users only. **OSPFMON** is intended to be used as a tool for debugging gateways, not for network management. SNMP is the preferred network management protocol. GateD responds only to **OSPFMON** requests issued on the same subnet on which GateD is running.

OSPFMON -x *subsysid filename*

subsysid Four character MVS subsystem ID of the Cisco IOS for S/390 job or started task

Default: ACSS

filename Name of a file that has a list of gateways

Example

The following is a sample of a list of gateways:

```
138.42.181.50 gateway1.company.com [password]
138.42.171.64 gateway2.company.com
138.42.181.200 gateway3.company.com
138.42.171.220 gateway4.company.com
138.42.181.3 gateway5.company.com
138.42.224.4 gateway6.company.com
```

The optional *password* is specified by the monitorauthkey sub-parameter of the ospf parameter of the GateD configuration file.

Local Commands

Enter **OSPFMON** for the desired subsystem ID. These commands are supported:

d	Show configured destinations
h	Show history
x	Exit
@ *remote_command*	Use last destination
@*dest_index remote_command*	Use configured destination index
F *filename*	Write monitor information to filename
S	monitor information to stdout

PING

Use the **PING** TSO diagnostic command to determine if a host is active on the network. **PING** sends an ICMP ECHO_REQUEST packet to network hosts to elicit an ICMP ECHO_RESPONSE from the specified host or network gateway. If the host responds, **PING** replies that the host is alive and then exits. Otherwise, after the timeout expires, **PING** replies that there was no answer from the host.

If a *count* is not specified, **PING** continues to try until it is stopped.

Note **PING** requires the SAS/C Transient Library modules that are supplied in the LOAD/SASLOAD data set. This library is required in either your LOGON procedure or BATCH job STEPLIB DD.

Source for **PING** and many BSD socket applications can be obtained via anonymous FTP from Internet host gatekeeper.dec.com.

If, after invocation, **PING** waits too long for a reply, press **PA1** to interrupt its execution (this may produce a dump of system related errors).

PING [-dnqv] [-c *count*] **[-i** *wait*] **[-l** *preload*] **[-p** *pattern*] **[-s** *packetsize*] **[-x** *subsysid*]

host

Syntax Description

-d	Set the SO_DEBUG socket option
-n	Show network addresses as numbers (**PING** normally displays addresses as host names)
-q	Run **PING** in quiet mode
-v	Verbose output; list any ICMP packets received, other than ICMP_RESPONSE
-c *count*	Specifies the number of requests to be sent
-i *wait*	Specifies the interval between successive transmissions Default: 1 second
-l preload	Specifies the count of pings initially sent
-p *pattern*	Fill the buffer with pattern
-s *packetsize*	Specifies the packet size to be sent Default: 64 bytes
-x *subsysid*	Specifies the subsystem ID of the Cisco IOS for S/390 job or started task; *subsysid* is the four-character MVS subsystem ID Default: ACSS
host	Specifies the host name or IP address

RIPQUERY

The **RIPQUERY** TSO diagnostic command can request all routes known by a RIP gateway by sending a RIP request or **POLL** command. The routing information in any routing packets returned is displayed numerically and symbolically.

Note The **RIPQUERY** tool executes as a TSO command processor. All messages are written with TPUTs, which restricts its use to interactive TSO users. The **RIPQUERY** tool is for debugging gateways, not for network management. SNMP is the preferred network management protocol.

By default, **RIPQUERY** uses the **RIP POLL** command for this version of GateD. The **RIP POLL** command is preferable to the **RIP REQUEST** command as it is not subject to Split Horizon and/or Poisoned Reverse. See the RIP RFC for more information.

RIPQUERY -x *subsysid* **[-a** *password***] [-d] [-n] [-p] [-r] [-v] [-1] [2] [-w** *time***]** *routers***...**

Syntax Description

-x *subsysid* Four character MVS subsystem ID of the Cisco IOS for S/390 job or started task.

Default: ACSS

-a *password* Authentication password to use for queries. If specified, an authentication type of SIMPLE will be used; otherwise the default is an authentication type of NONE. Authentication fields in incoming packets will be displayed but not validated.

-n Prevents the address of the responding host from being looked up to find the symbolic name.

-p The **RIP POLL** command requests information from the routing table. This is the default for some versions of GateD. If there is no response to **RIP POLL**, try the **RIP REQUEST** command. GateD responds to a **POLL** command with all the routes learned via RIP.

-r The **RIP REQUEST** command requests information from the gateway's routing table. All gateways should support **RIP REQUEST**. If there is no response to the **RIP REQUEST** command, try the **RIP POLL** command. GateD responds to a **REQUEST** command with all routes currently announced on the specified interface. For systems based on BSD 4.3 Reno or earlier, responses to **RIP REQUEST**s contain information about the interface used to send the reply. To obtain information about a particular host, run **RIPQUERY** on that host.

-v Displays version information about **RIPQUERY** before querying the gateways.

-1 Sends the query as a version 1 packet.

-2 Sends the query as a version 2 packet (default).

-w *time* Specifies the time in seconds to wait for the initial response from a gateway.

Default: 5 seconds

routers Router name list.

RPCINFO

The **RCPINFO** command queries an RPC server to obtain information about program names and program numbers for the specified host.

The **RPCINFO** command is supplied with Cisco IOS for S/390 as a TSO command processor for use on the MVS system. Its usage is similar to the **rpcinfo** command on UNIX systems. No parameters are positional or case sensitive.

Read *Cisco IOS for S/390 Unprefixed Messages and Codes* for information about **RPCINFO** error messages.

RPCINFO [PROTO (*protocol***)] [HOST(***host_name***)] [PROG(***prog_num***)]**

[VERS(*vers_num***)] [PORT(***port***)] [SYSID(***subsystem_id***)]**

Syntax Description

PROTO(*protocol*)	Specifies the protocol to use, either UDP or TCP.
HOST(*host_name*)	Specifies the name of the host where the service is resident.
PROG(*prog_num*)	Specifies the RPC program number in decimal.
VERS(*vers_num*)	Specifies the version number of the RPC program.
PORT(*port_num*)	Specifies the UDP or TCP port where the selected service is registered.
SYSID(*subsystem_id*)	Specifies the subsystem ID of Cisco IOS for S/390. Default: ACSS

Examples

- Use this **RPCINFO** command to dump the portmapper registration tables:

  RPCINFO PROTO(*protocol***) HOST(***host_name***) PROG(***prog_num***)**

  [VERS(*vers_num***)] [PORT(***port_num***)]**

  [SYSID(*subsystem_id***)]**

  Use this command to ping an RPC service:

  RPCINFO HOST(*host_name***) [SYSID(***subsystem_id***)]**

  Use this command to delete an RPC service registration with the LOCAL portmapper:

  RPCINFO DELETE PROG(*prog_num***) VERS(***vers_num***) SYSID(***subsystem_id***)**

- The following is a sample command and output for the **RPC HOST** command to list the services registered on MVS host zeus:

RPCINFO HOST(ZEUS)

```
RPC105I  Thu  Feb  07 10:47:10   1991  number  vers.  protocol  port  name
RPC106I  Thu  Feb  07 10:47:10   1991  100000   2     TCP       111   PORTMAPPER
RPC106I  Thu  Feb  07 10:47:10   1991  100000   2     UDP       111   PORTMAPPER
RPC106I  Thu  Feb  07 10:47:10   1991  100059   1     UDP       4098
RPC106I  Thu  Feb  07 10:47:10   1991  100044   1     UDP       4099
RPC106I  Thu  Feb  07 10:47:10   1991  100005   1     UDP       4100  MOUNTD
RPC106I  Thu  Feb  07 10:47:10   1991  150001   1     UDP       4101
RPC106I  Thu  Feb  07 10:47:10   1991  100003   2     UDP       2049  NFS
```

TCPEEP

TCPEEP is a TSO command that invokes the Cisco IOS for S/390 packet trace program to diagnose remote host communication problems. The TCPEEP real-time trace consists of selected network packet traffic to and from a local host. The TCPEEP command recognizes LNI level traffic and most of the IP-based higher level protocols.

TCPEEP will create a NO WRAP Component Trace Instance and display the output on a TSO terminal or direct it to a dynamically allocated SYSOUT data set. Optionally, it can stop any Component Trace Instance or modify an existing Component Trace Instance or view an existing Component Trace Instance.

Note TCPEEP runs only when the Cisco IOS for S/390 and TRACE address spaces are active.

User Interface

TCPEEP can be run as a TSO command, either from TSO or as a batch TSO.

The JCL to run TCPEEP as a batch job is in SAMP member TCPEEP. Cisco IOS for S/390 TRACE must be up and running before submitting a batch job for TCPEEP.

The following is a sample JCL for running TCPEEP in batch.

```
//TCPEEP JOB (TCPEEP),'TCPEEP', CLASS=A,MSGCLASS=X
//*
//* Sample JCL to run TCPEEP in batch.
//*
//* Update “trgindx" to reflect your library naming convention
//*
//* Note: The Cisco IOS for S/390 and TRACE address spaces must be running.
//*
//*TCPEEP     EXEC PGM=IKJEFT01,DYNAMNBR=50,REGION=4M
//STEPLIB     DD DISP=SHR,DSN=trgindx.LINK
//SYSPRINT    DD SYSOUT=*
//SYSTSPRT    DD SYSOUT=*,LRECL=132,BLKSIZE=132,RECFM=FB
//SYSIN       DD DUMMY
//*
//SYSTSIN     DD *
//*
 TCPEEP TRCSSID(ACTR) GROUPS(NETIF) +
        BUFFTIME(5)
/*
```

To stop the TCPEEP batch job, issue the MVS **STOP** command (for example, **P** *jobname*).

Trace Data Collected

The trace facility collects the following type of trace data:

- From an application
- TCP/IP address space

Simultaneous tracing of various trace types can be fed into a single trace output.

Note If data is to be collected via the external writer to an external data set, you must use the MVS TRACE command. Read External Writer for more information.

Viewing Trace Data

TCPEEP collects real time data and by default writes to SYSTSPRT. Optionally, it can write to a dynamically allocated SYSOUT data set.

Trace Operation

Each occurrence of the **TCPEEP** command varies according to the type of environment.

To stop a trace, press the terminal **ATTENTION** key and enter **H** at the prompt (null entry will allow trace to continue).

Caution **TCPEEP** should be installed in a protected library only. It can be used to display all network traffic through Cisco IOS for S/390, including user IDs and passwords.

TCPEEP Syntax

This section describes the **TCPEEP** syntax and describes its parameters.

TCPEEP [ASID(*asid***,...)] [BUFFERS (***size***,** *number***)] [BUFFTIME (***time_out***)]**
[DATASIZE (*record_size***)] [DNRSSID (***ssid***)] [FORMAT(***format_options***)]**
[FULL | SUMMARY] [GROUPS((*group* **[,'***filter***')...)] [HALT]**
[INSTANCE (*instance_ID***)] [JOBNAME (***jobname***,...)] [NOHEADER]**
[PEEK (*limit***)] [SYSOUT (***class***)] [TRACESIZE (***num_records)***]**
[TRCSSID (*ssid***)]**

Syntax Description

ASID
(*asid*,...)

Specifies the address space identifiers (ASIDs) of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas. An empty ASID list, ASID=(), turns off filtering by address spaces. In the ASID parameter, list all address spaces to be traced. Address spaces for previous traces are not traced unless listed.

Default: None

BUFFERS
(*size*,*number*)

Specifies the size of the trace buffers in kilobytes and/or the number of buffers. *size* may be a value between 64 and 1024 with the default being 256. *num* may be a value between 2 and 128 with the default being 4. BUFFERS is optional and may only be specified when creating a new trace instance. If specified when modifying a trace instance, it is ignored.

Range: *size*: 64 - 1024, *number:* 2 - 128

Default: (256, 4)

BUFFTIME

(*time_out*)

Specifies the buffer time out interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated. BUFFTIME is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

This parameter can be used to force a buffer switch so you don’t have to wait for the entire buffer to fill to see trace data.

Range: 0 - 99999

Default: 10

DATASIZE

(*record_size*)

Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated. DATASIZE is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored. If the specified maximum size exceeds the largest supported trace record size (64K less control headers), then the specification has no effect.

Default: None

DNRSSID

(*ssid*)

Specifies the subsystem ID of the address space to be used for name resolution by the GROUPS filtering process. Specification is ignored if GROUPS with filter parameters is not specified.

Default: ACSS

FORMAT

(*format_options*)

The format of the collected data.

Formatted data can be specified as follows:

- APCB—specifies the APCB should be formatted
- APCBHEX—Dump the APCB in hex
- ARP—selects ARP level trace
- ARPHEX—selects ARP level trace but dumps the ARP header in hex
- ASCII(*len*)—same as DATA(*len*) but uses an ASCII only translate table
- DATA(*len*)—displays the actual data transmitted. *len* is optional, the default being to display all the data. For example, FORMAT(DATA). Specifying FORMAT(DATA(16)) would display at most 16 (X’10’) bytes of data. This uses a mixed ASCII and EBCDIC translate table.
- DEVICE—display device/interface name
- EBCDIC(*len*)—same as DATA(len) but uses an EBCDIC only translate table
- IP—selects IP level trace
- IPHEX—selects IP level trace but dumps the IP header in hex
- ICMP—selects ICMP level trace
- ICMPHEX—selects ICMP level trace but dumps the ICMP header in hex
- LNI—selects the LNI trace (all LNI traffic is traced)
- TCP—selects TCP level trace
- TCPHEX—selects TCP level trace but dumps the TCP header in hex
- TPL—specifies the TPL should be formatted
- TPLHEX—dump the TPL in hex
- UDP—selects UDP level trace
- UDPHEX—selects UDP level trace but dumps the UDP header in hex

Format options for NETIF are ARP, ARPHEX, DATA(*nn*) | EBCDIC(*nn*) | ASCII(*nn*), DEVICE, IP, IPHEX, LNI, TCP, TCPHEX, UDP, UDPHEX,.

NETIF Defaults: ARP, IP, ICMP, LNI, TCP, UDP

Format options for TLI are APCB, APCBHEX, TPL, and TPLHEX.

TLI Defaults: APCB and TPL

FULL | SUMMARY

Show all or a summary of the data.

Default: SUMMARY

GROUPS
((*group*
[, '*filter*'])
...)

Trace group or groups for which data is collected and optionally a filter parameter for each group. Use GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

group - selects the type of data to collect: NETIF or TLI.

- NETIF is a network interface that includes the following *filter* options:
  - — HOST(*host*,...,*host*)—up to 16 IP HOST addresses (names)
  - — PORT(*port*,...,*port*)—up to 16 port numbers (names)
  - — PROTOCOL—a single protocol (for example, UDP)
  - — TYPE(*type*) for NETIF can be one of the following:

    TYPE(IP)—if datagram is an IP packet, then keep it

    TYPE(ARP)—if datagram is an ARP packet, then keep it

    TYPE(ALL)—keep all datagrams (same as not specifying the filter)

  There are no defaults for *filters*; everything is collected by default.
- TLI is a collection of trace points in the stack for processing TLI activity with the following *filter* options:
  - — HOST(*host*,...,*host*)—up to 16 IP HOST addresses (names)
  - — PORT(*port*,...,*port*)—up to 16 port numbers (names)
  - — UASID(*asid*,...,*asid*)—takes one to 16 asids, which refer to jobs using the TLI function
  - — USER(*jobname*,..,*jobname*)—takes one to 16 jobnames, which refer to jobs using the TLI function

Refer to the section Useful TCPEEP Commands for usage of the GROUPS parameter.

Default: NETIF

HALT

Stop a component trace instance.

INSTANCE
(*instance_id*)

By specifying the *instance_id* returned from **TCPEEP** or the MVS **TRACE CT** command, you can select a trace instance to display.

JOBNAME
(*jobname*,....)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

NOHEADER

Do not display the header information from the trace entry. This option helps limit the output. Without it one line will print describing the trace entry even if no other information about that entry displays.

PEEK
(*limit*)

Number of trace records to view. Use to view an existing trace. PEEK or PEEK(0) implies no limit; PEEK(*n*) traces only *n* trace records.

Range: 0 - no limit.

Default: 0

SYSOUT (*class*)

Output sent to a dynamically allocated SYSOUT data set.

By default, the output for **TCPEEP** writes to SYSTSPRT.

Default: X.

TRACESIZE (*num_records*)

Maximum number of trace records to be recorded. If not specified, there is no limit to the number of records recorded. TRACESIZE is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

TRCSSID (*ssid*)

ID of the trace address space.

Default: ACTR

Useful TCPEEP Commands

This section lists the keywords for each function listed.

Creating a NOWRAP Trace

Use the following command to initiate a NO WRAP trace instance:

```
TCPEEP TRCSSID(trcsubsysid) DNRSSID(dnrsubsysid) GROUPS(group)      +
  JOBNAME(jobname) ASID(asid) DATASIZE(size) TRACESIZE(trcsize)      +
  BUFFERS(bufname) BUFFTIME(buftime) FORMAT(format_options)      +
  SYSOUT(class)
```

Stopping Trace

Use this command to stop any trace instance:

```
TCPEEP INSTANCE(inst) HALT TRCSSID(trcsubsysid)
```

Modify Existing Trace

Use this command to modify an existing trace instance:

```
TCPEEP INSTANCE(inst) TRCSSID(trcsubsysid) DNRSSID(dnrsubsysid)         +
  GROUPS (group) JOBNAME(jobname) ASID(asid)   PEEK                 +
  FORMAT(format_options) SYSOUT(class)
```

Regardless of how the trace is started you will view it LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

View Existing Trace

Use this command to view an existing trace instance:

```
TCPEEP INSTANCE(inst) TRCSSID(trcsubsysid) PEEK                          +
  FORMAT(format_options) SYSOUT
```

Regardless of how the trace is started you will view it LIFO for the specified number of records (PEEK or PEEK(0) implies no limit).

Specifying GROUPS

This example shows how to specify the GROUPS for TCPEEP.

```
TCPEEP GROUPS ( (NETIF, ’HOST (host1, host2), PORT (ECHO) ’),               +
   (NETIF, ’HOST (host3) ’))
```

This example allows you to specify NETIF without additional filtering.

```
TCPEEP GROUPS (NETIF)
```

To specify NETIF and PROTOCOL(UDP):

```
TCPEEP GROUPS ((NETIF, 'PROTOCOL (UDP)'))
```

TCPEEP Examples

This section shows the output from several **TCPEEP** commands.

Default TCPEEP Command

This sample shows the output for a TCPEEP command with no filtering.

```
3172    00010100 03/26 12:47:13.246650 INPUT
3172 Control:Offset=0, Next=20, Data Len=20, Adapter=0
  CMD= Shutdown_ Flag= ICP-Init
------------------------------------------------------------
3172    00010200 03/26 12:47:13.247232 OUTPUT
FDDI:Offset=0, Next=73, Data Len=73, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
IP:Src=192.120.80.9, Dst=9.12.120.110, TotLen=48, Prot=6, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
TCP:12:47:13.247232 192.120.80.9:1234 --> 9.12.120.110:9
  Seq:1,9(8) Ack:2 Psh
  HdrLen=20, Win=2048(2050), Cksum=9BAC, Flags=08
-----------------------------------------------------------
3172    00010100 03/26 12:47:13.247546 INPUT
FDDI:Offset=0, Next=61, Data Len=61, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
IP:Src=192.120.80.8, Dst=9.12.120.110, TotLen=36, Prot=17, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
UDP:Src Port=5678, Dst Port=9, Len=16, Cksum=F35A
-----------------------------------------------------------
3172    00010100 03/26 12:47:13.247706 INPUT
FDDI:Offset=0, Next=61, Data Len=61, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
IP:Src=192.120.80.8, Dst=9.12.120.110, TotLen=36, Prot=1, Id=20000
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
ICMP:Echo Response,_ Cksum=5D52
  Id=04D2, Sequence Number=5678
-----------------------------------------------------------
3172    00010100 03/26 12:47:13.248779 INPUT
Token Ring:Offset=0, Next=316, Data Len=316, Adapter=1
  Access Control=10, Frame Control=40
  Destination=FF.FF.FF.FF.FF.FF, Source=92.34.56.78.9A.BC
  Control Byte 1=06, Control Byte 2=B0
  Segments:1234 5678
  Dsap=01, Ssap=02, Ctl=03
  ProtType=0800 (IP)
IP:Src=192.120.80.9, Dst=9.12.120.110, TotLen=281, Prot=6, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
TCP:12:47:13.248779 9.12.120.110:9 <-- 192.120.80.9:1234
  Ack:2 Seq:1,242(241) Urg Psh
  HdrLen=20, Win=2048(2050), UrgPtr=2(3), Cksum=9BAC, Flags=28
------------------------------------------------------------
```

LNI Level Trace

This example shows the LNI level trace output.

```
3172     00010100 03/26 13:00:53.774527 INPUT
3172 Control:Offset=0, Next=20, Data Len=20, Adapter=0
  CMD= Shutdown_ Flag= ICP-Init
------------------------------------------------------------
3172     00010200 03/26 13:00:53.774881 OUTPUT
FDDI:Offset=0, Next=73, Data Len=73, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
------------------------------------------------------------
3172     00010100 03/26 13:00:53.775035 INPUT
FDDI:Offset=0, Next=61, Data Len=61, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC_
  ProtType=0800 (IP)
------------------------------------------------------------
3172     00010100 03/26 13:00:53.775967 INPUT
Token Ring:Offset=0, Next=316, Data Len=316, Adapter=1
  Access Control=10, Frame Control=40
  Destination=FF.FF.FF.FF.FF.FF, Source=92.34.56.78.9A.BC_
  Control Byte 1=06, Control Byte 2=B0
  Segments:1234 5678
  Dsap=01, Ssap=02, Ctl=03
  ProtType=0800 (IP)
------------------------------------------------------------
CETI    00010103 03/26 13:00:53.778563 INPUT
CETI Ethernet:Flags=00, Data Len=50
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
------------------------------------------------------------
```

DATA Output

This sample shows the output for viewing the data.

```
3172    00010200 03/26 13:03:50.826067 OUTPUT
FDDI:Offset=0, Next=73, Data Len=73, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
IP:Src=192.120.80.9, Dst=9.12.120.110, TotLen=48, Prot=6, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
TCP:13:03:50.826067 192.120.80.9:1234 --> 9.12.120.110:9
  Seq:1,9(8) Ack:2 Psh
  HdrLen=20, Win=2048(2050), Cksum=9BAC, Flags=08
DATA +0000 E3C3D740 C4C1E3C1                *TCP@DATA      *
-----------------------------------------------------------
3172    00010100 03/26 13:03:50.826222 INPUT
FDDI:Offset=0, Next=61, Data Len=61, Adapter=2
  Destination=FF.FF.FF.FF.FF.FF, Source=52.34.56.78.9A.BC
  ProtType=0800 (IP)
IP:Src=192.120.80.8, Dst=9.12.120.110, TotLen=36, Prot=17, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
UDP:Src Port=5678, Dst Port=9, Len=16, Cksum=F35A
DATA +0000 55445020 64617461  *UDP data         *
-----------------------------------------------------------
3172    00010100 03/26 13:03:50.827670 INPUT
Token Ring:Offset=0, Next=316, Data Len=316, Adapter=1
  Access Control=10, Frame Control=40
  Destination=FF.FF.FF.FF.FF.FF, Source=92.34.56.78.9A.BC
  Control Byte 1=06, Control Byte 2=B0
  Segments:1234 5678
  Dsap=01, Ssap=02, Ctl=03
  ProtType=0800 (IP)
IP:Src=192.120.80.9, Dst=9.12.120.110, TotLen=281, Prot=6, Id=38827
  Vers=4, Hdr Len=20, TOS=0, TTL=60, Offset=0(0), Cksum=68C0
TCP:13:03:50.827670 9.12.120.110:9 <-- 192.120.80.9:1234
  Ack:2 Seq:1,242(241) Urg Psh
  HdrLen=20, Win=2048(2050), UrgPtr=2(3), Cksum=9BAC, Flags=28
DATA +0000 54686973 20697320 41534349 49203031 *This is ASCII 01*
     +0010 00000000 00000000 00000000 00000000 *................*
     +0020 00000000 00000000 00000000 00000000 *................*
     +0030 00000000 00000000 00000000 00000000 *................*
     +0040 00000000 00000000 00000000 00000000 *................*
     +0050 00000000 00000000 00000000 00000000 *................*
     +0060 00000000 00000000 00000000 00000000 *................*
     +0070 00000000 00000000 00000000 00000000 *................*
     +0080 00000000 00000000 00000000 00000000 *................*
     +0090 00000000 00000000 00000000 00000000 *................*
     +00A0 00000000 00000000 00000000 00000000 *................*
     +00B0 00000000 00000000 00000000 00000000 *................*
     +00C0 00000000 00000000 00000000 00000000 *................*
     +00D0 00000000 00000000 00000000 00000000 *................*
     +00E0 00E38889 A24089A2 40C5C2C3 C4C9C340 *.This@is@EBCDIC@*
     +00F0 F0                                  *0               *
-----------------------------------------------------------
```

ICMP Format

This example shows the output formatted for ICMP filtering.

```
3172     00010100 03/26 13:06:16.977131 INPUT
ICMP:Destination Unreachable,_ Cksum=5D52
  Code=1 (Hos                    )
  IP:Source=192.120.99.4, Dest=9.12.99.5, Proto=6, Tot Len=200
  Vers=4, Hdr Len=36, TOS=0, TTL=60, Id=1234, Offset=0, Cksum=68C0
-----------------------------------------------------------
3172____ 00010100 03/26 13:06:16.978076 INPUT
ICMP:Source Quench,  Cksum=5D52
  IP:Source=192.120.99.4, Dest=9.12.99.5, Proto=6, Tot Len=200
  Vers=4, Hdr Len=36, TOS=0, TTL=60, Id=1234, Offset=0, Cksum=68C0
-----------------------------------------------------------
```

TRACE

Cisco IOS for S/390 contains enhancements to the IBM MVS Component Trace facility to provide a method of collecting TCP/IP data and displaying it on a terminal or sending it to an external writer. Additional JCLs are required to enable the Component Trace enhancements. To use Component Trace as a TSO command, read TCPEEP.

MVS Component Trace

MVS Component Trace is a diagnostic aid used to trace the action of certain system components and third party components that define themselves to Component Trace. The TRACE operator command is used to start, stop and control the component trace. For more information on the MVS TRACE command, see the IBM MVS System Commands.

Trace Address Space

Component Trace is defined in its own address space and collects trace data for trace points defined in other address spaces. Likewise, the other address spaces must identify the Component Trace address space for the data collection. Multiple occurrences of Component Trace can be active at the same time, each with a unique subsystem ID.

For each address space, IJTCFGxx in the PARM member defines the subsystem ID. The definition is specified using the TRACENAME keyword on the IFSPARM statement as follows:

IFSPARM PROMPT VMCFNAME(VMCF) TRACENAME(ACTR) NOPROMPT

In the example, the trace data in the address space is directed to the Component Trace address space with a ACTR subsystem ID.

Note If the Trace Address Space is brought up after the TCP stack, there may be a delay of 2 minutes before events are traced.

Exit

MVS Component Trace requires an exit to communicate with the tracing component. The exit, T03PTRSS, must reside in LPALIB or the Link List; it is distributed in the LINK library.

The Component Trace subsystem ID is defined with the UNIQ= keyword which is part of the parameter string specified on the JCL EXEC PARM= statement. Normally the value for the UNIQ= keyword is specified on the SSN= keyword of the PROC statement as shown below:

```
//ICSTRACE PROC TRGINDX=trgindx, TARGET LIBRARIES DSN INDEX
//     SSN=ACTR,            DFLT SUBSYSTEM NAME
//     SOUT='X',            CHOOSE A HOLD NONPURGE SYSOUT CLASS
//     CMND=STARTTR         DFLT STARTUP COMMAND SCRIPT NAME
//*
//TRACE EXEC PGM=IFSSTART,REGION=6144K,TIME=1440,
// PARM='IFSINIT,UNIQ=&SSN,PRFX=T03,SOUT=&SOUT,CMND=&CMND'
```

External Writer

The collected trace data is written to DASD or TAPE using an External Writer. A suitable External Writer Cataloged Procedure for use with Component Trace may already be defined on your system. For more information on defining an External Writer used with Component Trace, see the IBM publication - MVS Diagnosis: Tools and Service Aids.

TRACE Command

The **TRACE** command starts, stops, modifies, or displays the status of a system trace, master trace, or component trace.

The **TRACE** command is a standard MVS operator command.

```
TRACE   [CT {[, ON | OFF] [, COMP=name] [, PARM=mem]}]
        [CT {[, WTRSTART=mem_name[,WRAP | NOWRAP]]}]
        [CT {, WTRSTOP=job_name}]]
```

Syntax Description

ON Turns on tracing for a component if the component trace is currently off. If the component trace is on and can be changed, this changes the trace options.

OFF Turn off tracing for the component. If the component is connected to an external writer, the trace is implicitly disconnected from the writer.

COMP=*name* Identifies the component trace with the subsystem ID for the Trace address space. This is required for each **TRACE** command.

PARM=*mem* Identifies a member of SYS1.PARM or a data set in the system parmlib concatenation containing the parameters used for tracing. Using a parmlib member allows the operator to initiate the trace, change it, or stop it without a message prompting for parameters. Parameters specified on the **TRACE** command override the options specified in the parmlib member. The parameters are described in the section TRACE Command Reply.

WTRSTART= *mem_name* Identifies the member containing the JCL to invoke an external writer and opens the data sets used by the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job. After starting the external writer, use the WTR parameter to connect the component trace to the external writer.

WRAP | NOWRAP Using the **NOWRAP** parameter, instructs the system to stop writing data to a data set when the data set is full. With the WRAP parameter, when the data set or group of data sets is full, new data over-writes the oldest data at the start of the data set or the start of the first data set.

If the **WTRSTART** parameter on the **TRACE CT** command specifies **NOWRAP**, the system uses the primary and secondary extents of the data set or sets. If the **WTRSTART** parameter specifies **WRAP** or omits the parameter, the system uses only the primary extent or extents.

WTRSTOP= *job_name* Disconnects the external writer from the component trace and closes the data sets used by the external writer. The jobname is the member name if the source JCL is a procedure, or a Job name if defined on a JOB statement within the source JCL.

Before stopping the external writer, turn the component trace off with **TRACE CT,OFF** or disconnect the external writer with **WTR=DISCONNECT**.

TRACE Command Reply

In response to a **TRACE CT,ON** command without the **PARM** parameter, the system prompts you to specify the component trace options. Use the **REPLY** command to respond.

R *id***[,ASID=(***nnnn* **[,***nnnn***]...)]**

[,CONT |,END]

[,JOBNAME=(*name* **[,***name***]...)]**

[,OPTIONS=(*option* **[,***option***]...)]**

[,WTR={*mem_name* **| DISCONNECT}]**

Syntax Description

id

Use the same identification number (0-9999) from the message to identify the reply.

ASID=

(*nnnn* **[,** *nnnn***]...**)

Specifies the address space identifiers (ASIDs) of address spaces used as a filter for tracing. Events in the ASIDs are recorded by the component trace.

The parameter contains a list of 0 to 16 hexadecimal ASIDs separated by commas. An empty ASID list, ASID=(), turns off filtering by address spaces. In the ASID parameter, list all address spaces to be traced. Address spaces for previous traces are not traced unless listed.

CONT or **END**

The CONT parameter continues the reply on another line. The system issues another reply message. You can then continue the reply and repeat any parameters on the continuation line, except END. Repeated parameters are strung together, they do not overlay each other. The END parameter identifies the end of the REPLY.

CONT or END must be the last parameter on the input line.

JOBNAME=

(*name* **[,** *name***] ...**)

Names of jobs used as filters for tracing. Events in these jobs are recorded by the component trace.

The parameter contains a list of 0 to 16 job names separated by commas. An empty job list, JOBNAME=(), turns off filtering by jobs. In the JOBNAME parameter, list all jobs to be traced. Jobs specified for previous traces are not traced unless listed.

OPTIONS=(*option* **[,** *option***] ...**)

Specifies the component trace options described in the section TRACE Command Reply Options.

WTR= *mem_name* | DISCONNECT

The *membername* identifies the member containing the source JCL that invokes the external writer. The member must be a SYS1.PROCLIB cataloged procedure or a job. The membername in the WTR parameter must match the membername in the **TRACE CT,WTRSTART** command.

WTR=DISCONNECT disconnects the external writer. The component continues tracing and placing the trace records in the address-space buffer, but stops passing trace records to the external writer.

You must also specify a **TRACE CT**,**WTRSTART** or **TRACE CT,WTRSTOP** command to start or stop the writer.

TRACE Command Reply Options

Use the Options parameter in response to the Reply prompt.'

OPTIONS= [BUFFERS(*size***,***num***)] [BUFFTIME(***time***)] [DATASIZE(***dsize***)]**
[DNRSSID(*ssid*)] [GROUPS((*group*[, '*filter*'])...)] [HALT]
[INSTANCE(*inst*)] [STATUS] [TRACESIZE(*tsize***)] [WRAP]**
[NOWRAP]

Syntax Description:

BUFFERS (*size*,*num*) Size of the trace buffers in kilobytes and/or the number of buffers. *size* may be a value between 64 and 1024 with the default being 256. *num* may be a value between 2 and 128 with the default being 4. BUFFERS can only be specified for a new Trace Instance. BUFFERS is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

BUFFTIME(*time*) Buffer timeout interval in seconds. At the end of each interval, if the current buffer contains data but is not full, a buffer flush operation is initiated. BUFFTIME is optional and is considered only when creating a trace instance.

This parameter can be used to force a buffer switch so you don't have to wait for the entire buffer to fill to see trace data.

Range: 0 - 99999

Default: 10

DATASIZE(*dsize*) Specifies the maximum size of a trace record in kilobytes. Trace records that exceed the specified value are truncated. DATASIZE is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored. If the specified maximum size exceeds the largest supported trace record size (64K less control headers), then the specification has no effect.

Default: Any size record will be recorded (up to the maximum IBM limit of 64K less control headers.)

DNRSSID(*ssid*) Specifies the subsystem ID of the address space to be used for name resolution by the GROUPS filtering process. Specification is ignored if GROUPS with filter parameters is not specified.

Default: ACSS

GROUPS(
(*group* [, '*filter*'])...
)

Trace group or groups for which data is collected and optionally a filter parameter for each group. Use GROUPS parameter to limit the amount of data collected. Used with INSTANCE to modify an existing trace instance, either adds a new group to the trace instance or replaces an existing group for the trace instance. Once added, a group cannot be removed from the active trace instance.

groupid - selects the type of data to collect: NETIF or TLI.

- NETIF is a network interface with the following *filter* options:
  - — HOST—up to 16 IP HOST addresses (names), for TLI group only
  - — PORT—up to 16 port numbers (names), for TLI group only
  - — PROTOCOL—a single protocol (for example, UDP)
  - — TYPE(*type*) for NETIF can be one of the following:

    TYPE(IP)—if datagram is an IP packet, then keep it

    TYPE(ARP)—if datagram is an ARP packet, then keep it

    TYPE(ALL)—keep all datagrams (same as not specifying the filter)

  There are no defaults for *filters*; everything is collected by default.
- TLI is a collection of trace points in the stack for processing TLI activity with the following *filter* options:
  - — HOST—up to 16 IP HOST addresses (names)
  - — PORT—up to 16 port numbers (names)
  - — UASID(*asid*,...,*asid*)—takes one to 16 asids, which refer to jobs using the TLI function
  - — USER(*jobname*,..,*jobname*)—takes one to 16 jobnames, which refer to jobs using the TLI function

Refer to the section Useful TCPEEP Commands for usage of the GROUPS parameter.

Default: NETIF

HALT

HALT specifies that the trace instance identified by the INSTANCE keyword is to be stopped. If HALT is specified, INSTANCE must be specified. If the INSTANCE is connected to an external writer, WTR=DISCONNECT must also be specified.

INSTANCE (*inst*)

Modify or stop a trace instance. *inst* is the component trace address space ID. A new trace instance is created if *inst* is not specified.

STATUS

Display all active trace instances. If STATUS is specified, any other keywords are ignored.

TRACESIZE(*tsize*)

Maximum number of trace records to be recorded. If not specified, there is no limit to the number of records recorded. TRACESIZE is optional and may only be specified when creating a trace instance. If specified when modifying a trace instance, it is ignored.

WRAP

Create an in-memory WRAP trace. Not valid with WTR=mem_name, nor can the instance be modified later to specify WTR=mem_name. This is the default if NOWRAP is not specified and WTR=mem_name is omitted.

NOWRAP Create a NOWRAP trace. This is the default if WTR=mem_name is specified. The external writer writes the buffers when they fill up and after being written are reused. This option is for the Trace Address space and should not be confused with WRAP|NOWRAP on the TRACE CT command, which is for the IBM writer.

Comparing TCPEEP and MVS TRACE Syntax

The syntax for commands using TCPEEP differ from those used with the MVS TRACE facility.

OPTIONS

To use MVS TRACE to specify NETIF and PROTOCOL(UDP), you need to specify:

OPTIONS = (GROUPS ((NETIF, ‘PROTOCOL (UDP)')))

This is the same as the TSO **TCPEEP** command:

TCPEEP GROUPS ((NETIF, 'PROTOCOL (UDP)'))

JOBNAME

To specify JOBNAME using the MVS TRACE command, the syntax is

JOBNAME = (*jobname1***, ...,** *jobname16***)**

When using **TCPEEP** the syntax is

TCPEEP JOBNAME (*jobname1***, ...,** *jobname16***)**

For more information about TCPEEP, read TCPEEP.

Trace Command Examples

The following examples demonstrate the control of the T03 Trace Facility using the **MVS TRACE** command. The examples assume a T03 Trace Facility is active and is using a Subsystem ID of ACTR.

Starting a Trace Instance

In this example a new trace instance is created. The instance is limited to tracing 10000 records for group id NETIF from address space TCP51.

TRACE CT,ON,COMP=ACTR

R *xx***,OPTIONS=(GROUPS(NETIF) TRACESIZE(10000)),JOBNAME=(TCP51),END**

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance

In this example an existing trace instance is modified. The trace instance is changed to include records for group id API.

TRACE CT,ON,COMP=ACTR

R *xx*,OPTIONS=(INSTANCE(1) GROUPS(TCP)),END

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Displaying Trace Status

In this example the status of the trace instances is displayed.

TRACE CT,ON,COMP=ACTR

R *xx*,OPTIONS=(STATUS),END

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR916I Instance(01) Active, records=23,745
T03TR916I Instance(02) Active, records=1,576
```

Stopping a Trace Instance

In this example an existing trace instance is stopped.

TRACE CT,ON,COMP=ACTR

R *xx*,OPTIONS=(INSTANCE(1) HALT),END

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping all Trace Instances

In this example all existing trace instances are stopped.

TRACE CT,OFF,COMP=ACTR

```
T03TR910I Trace shutdown successful Instance(01)
T03TR910I Trace shutdown successful Instance(02)
```

Starting an External Writer

In this example an external writer is started. T03XWTR is the name of a pre-defined started task.

TRACE CT,WTRSTART=T03XWTR

Starting a Trace Instance and connecting an External Writer

In this example a new trace instance is created and connected to an external writer. Only one instance can have an external writer connected.

TRACE CT,ON,COMP=ACTR

R *xx*,OPTIONS=(GROUPS(NETIF)),WTR=T03XWTR,END

Where *xx* ITT006A specifies the **TRACE CT** command operand. The resulting message is:

```
T03TR909I Trace start successful Instance(01)
```

Modifying a Trace Instance to connect an External Writer

In this example an existing trace instance is connected to an external writer. Only one instance can have an external writer connected.

TRACE CT,ON,COMP=ACTR

R *xx***,OPTIONS=(INSTANCE(1)),WTR=T03XWTR,END**

Where *xx* ITT006A SPECIFY OPERAND(S) FOR TRACE CT COMMAND. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Modifying a Trace Instance to disconnect an External Writer

In this example an existing trace instance is disconnected from an external writer.

TRACE CT,ON,COMP=ACTR

R *xx***,OPTIONS=(INSTANCE(1)),WTR=DISCONNECT,END**

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR911I Trace modify successful Instance(01)
```

Stopping a Trace Instance and Disconnecting an External Writer

In this example an existing trace instance is to be stopped and disconnected from an external writer.

TRACE CT,ON,COMP=ACTR

R *xx***,OPTIONS=(INSTANCE(1) HALT),WTR=DISCONNECT,END**

Where *xx* ITT006A specifies the TRACE CT command operand. The resulting message is:

```
T03TR910I Trace shutdown successful Instance(01)
```

Stopping an External Writer

In this example an external writer is stopped. The writer must be disconnected from a trace instance before the it can be stopped.

TRACE CT,WTRSTOP=T03XWTR

Processing Trace Data

In order to process the data collected with the TRACE command, use the **CTRACE** command of IPCS. It handles trace data that is in the MVS Component Trace Entry (CTE) format. By collecting trace data in this format, it is possible to use the facilities of IPCS to format the trace data. IPCS exit routines need to be written to locate trace records in dumps, filter trace records, and drive trace record formats.

TRACERT

The **TRACERT** (trace route) TSO diagnostic command prints the route that packets take to get to a network host. **TRACERT** utilizes the IP protocol time-to-live field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host. The only required parameter is the host name or IP address.

Note To use the **TRACERT** command, you must PROTOCOL(RAW) defined in your TCPCFGxx member. **TRACERT** requires the SAS/C Transient Library modules that are supplied in the LOAD/SASLOAD data set. This library is required in either your LOGON procedure or BATCH job STEPLIB DD.

TRACERT [-drv] [*data_size***] [-g** *gateway***]** *host* **[-m** *max_ttl***] [-p** *port#***] [-q** *nqueries***]**
[-s *src_addr***] [-n] [-t** *ts***] [-w** *wait***] [-x** *subsysid***]**

Syntax Description

-d Turns on socket debugging (SO_DEBUG).

-r Bypass normal routing tables and send directly to a host on an attached network. If the host is not on a directly attached network, an error is returned. Use to ping a local host through an interface that has no route through it (for example, after the interface was dropped by routed).

-v Specifies verbose output. Any ICMP packets that are received (other than TIME_EXCEEDED and UNREACHABLE) are listed.

data_size Specifies probe datagram length.

Default: 38 bytes

-g *gateway* Enables the IP LSRR (Loose Source Record Route) option in addition to the TTL tests. This is useful for asking how the host at IP address gateway reaches a particular target.

host Specifies the host name or IP address.

-m *max_ttl* Maximum time-to-live (number of hops) used in outgoing probe packets will be *max_ttl*.

Default: 30 hops

-p *port#* Base UDP port number used in probes will be *port#*. **TRACERT** assumes that nothing is listening on UDP ports base to (base+nhops-1) at the destination host (so an ICMP message PORT_UNREACHABLE will be returned to terminate the route tracing). If something is listening on a port in the default range, this option can be used to pick an unused port range.

Default: 33434

-q *nqueries* Specifies the number of probes to sent.

Default: 3

-s *src_addr* Specifies that *src_addr* be used as the source address in outgoing probe packets. *src_addr* must be specified as an IP number. On hosts with more than one IP address, this option can force the source address to be something other than the IP address of the interface the probe packet is sent on. If the IP address is not one of this machine's interface addresses, an error is returned and nothing is sent.

-n Specifies that hop addresses be printed numerically rather than symbolically and numerically. This saves a nameserver address-to-name lookup for each gateway found on the path.

-t *ts* Specifies the type of service in probe packets to be *ts*. This value must be a decimal integer in the range 0 to 255. Use this option to determine if different types of service result in different paths. Not all values of ts are legal or meaningful; see the IP specification for definitions. Useful values are -t 16 (low delay) and -t 8 (high throughput).

Default: 0

-w *wait* Sets the time to wait for a response to a probe to wait seconds.

-x *subsysid* Specifies the subsystem ID of the Cisco IOS for S/390 job or started task. *subsysid* is the four character MVS subsystem ID.

Default: ACSS

Usage Guidelines

To trace the route an IP packet travels to an internet host, **TRACERT** launches UDP probe packets with a small TTL, then listens for an ICMP time exceeded (TIME_EXCEEDED) reply from a gateway. Probes are started with a TTL of 1 and are incremented by 1 until an ICMP port unreachable message (UNREACHABLE) is returned, which means the host was reached, a maximum number of hops is exceeded, or timeout is exceeded. Three probes are sent for each TTL setting.

Example

The following example shows the output from a **TRACERT** command.

```
  1   helios.ee.lbl.gov  (128.3.112.1)  0 ms  0  ms  0  ms
  2   lilac-dmc.Berkeley.EDU  (128.32.216.1)  39  ms  19  ms  39  ms
```

A line is displayed showing the TTL, gateway address, and round trip time of each probe. If answers come from several gateways, the address of each system is displayed. If no response is received in the 3 second timeout period, an asterisk (*) is displayed for that probe.

The following list describes some possible notations:

! An exclamation point following the time indicates the TTL is less than or equal to 1

H Host unreachable received

N Network unreachable received

P Protocol unreachable received

Note **TRACERT** is useful for network testing, measurement, and management and should be used primarily for fault isolation. It can impose a load on the network and should not be used during normal operations or from automated scripts. Source for the UNIX program traceroute and many BSD socket applications can be obtained via anonymous FTP from Internet host gatekeeper.dec.com.