V5R1 APAR OA04304 Readme
IP Packet Trace Management and Formatting

Version 5 Release 1
V5R1 APAR OA04304 Readme
IP Packet Trace Management and Formatting

Version 5 Release 1
Before using this information and the product it supports, read the information in “Notices” on page 25.
## Contents

**About this Program Temporary Fix** .................................................. v
New in this PTF ................................................................................. v
Backward compatibility ................................................................. v

**Software requirements** ................................................................. 1

**Documentation Notes** ................................................................. 3
Enabling packet trace data ................................................................. 3
Controlling the collection of packet trace data ................................. 3
CNMSTYLE Statements ...................................................................... 4
INIT.PKTS ....................................................................................... 4
PKTS.STORAGE ............................................................................. 4

Commands ......................................................................................... 5
  FMTPACKT (NCCF, CNMFPFTC) .................................................... 6
  LIST (NCCF) ............................................................................... 7
  PKTS ........................................................................................... 8

Pipeline stages and syntax ............................................................... 12
  PIPE FMTPACKT ......................................................................... 13

Messages .......................................................................................... 18

**Contacting Customer Support** ...................................................... 23

**Notices** ......................................................................................... 25
Trademarks ....................................................................................... 27
About this Program Temporary Fix

This Readme document provides important information about Tivoli® NetView® for z/OS® Version 5 Release 1, for APAR OA04304. These notes are the most current information for the product and take precedence over all other documentation.

Please review these notes thoroughly before installing or using this product.

New in this PTF

This PTF enables IP packet trace management and formatting.

Backward compatibility

This PTF applies to NetView for z/OS Version 5 Release 1 only.
Software requirements

To collect packet trace information using the socket interface, you must be running z/OS with Communications Server V1R4 with PTFs applied for the following APARs:

- PQ77244
- PQ77837
- PQ77838
- PQ77840
Documentation Notes

The following sections provide information about the packet tracing and formatting functions provided in this PTF:

- “Enabling packet trace data”
- “CNMSTYLE Statements” on page 4
- “Controlling the collection of packet trace data”
- “Commands” on page 5
- “Pipeline stages and syntax” on page 12
- “Messages” on page 18

Enabling packet trace data

To collect packet trace information using the socket interface, enable the PKTTRCService service on the NETMONitor PROFILE.TCPIP statement in the z/OS Communications Server.

Controlling the collection of packet trace data

NetView initialization definitions are contained in DSIPARM member CNMSTYLE and its included members. To make changes to a CNMSTYLE statement, copy the statement to member CxxSTGEN and then make any necessary updates. For more information on NetView initialization and CNMSTYLE, refer to IBM Tivoli NetView for z/OS Installation: Getting Started.

To configure the management of TCP/IP packet trace data in the NetView product, perform the following steps:

- Copy the TOWER statement from CNMSTYLE to %INCLUDE member CxxSTGEN, and add the TCPIPCOLLECT tower.
- Add the following PKTS subtower to member CxxSTGEN:
  TOWER.TCPIPCOLLECT = PKTS
- Define an autotask to handle the collection of packet trace data for a stack. To do this automatically during NetView startup, add a FUNCTION.AUTOTASK statement to CxxSTGEN as follows:
  function.autotask.PKTS.stackname=taskname

  The stackname variable is the name of the TCP/IP stack, and the taskname variable is the name of the autotask responsible for collecting packet trace data for the stack.

  You can also define this autotask interactively using the PKTS DEFINE command. For more information, see “PKTS” on page 8.

- If you want to start packet trace data collection automatically during NetView startup, add the following statement to CxxSTGEN:
  INIT.PKTS = Yes

  You can also start data collection interactively using the PKTS START command. For more information, see “PKTS” on page 8.

- To specify the number of megabytes of data space storage allocated to packet trace data collected by the NetView program, add the following statement to CxxSTGEN:
PKTS.STORAGE.stackname=storg

Note: This statement applies only if collection of packet trace data is started automatically during NetView initialization using the INIT.PKTS statement. To specify storage allocation when starting data collection manually, use the STORAGE key word on the PKTS START command. (See "PKTS" on page 8 for more information.)

- Recycle the NetView program for the changes in member CxxSTGEN to take effect.

CNMSTYLE Statements

NetView initialization definitions are contained in DSIPARM member CNMSTYLE and its included members. To make changes to a CNMSTYLE statement, copy the statement to CxxSTGEN and then make any necessary updates. For more information on NetView initialization and CNMSTYLE, refer to IBM Tivoli NetView for z/OS Installation: Getting Started.

INIT.PKTS

The INIT.PKTS statement starts the IP packet trace management function during initialization of the NetView program.

The syntax for the INIT.PKTS statement is:

INIT.PKTS

Where:

NO
  Indicates not to start IP packet trace management.

YES
  Indicates to start IP packet trace management.

Usage Notes:

- You can start IP packet trace management at a later time by issuing the PKTS START command.

PKTS.STORAGE

The PKTS.STORAGE statement specifies the number of megabytes of data space storage allocated to packet trace data collected by the NetView program. The syntax of PKTS.STORAGE is as follows:

PKTS.STORAGE

Where:

stackname
  The name of a TCP/IP stack as defined by Communications Server.
The number of megabytes of data space storage to allocate. This value must be a positive or negative integer, followed by the letter M. A positive value indicates that the packet storage wraps. That is, when the storage area is full, arriving packets cause the oldest packets to be purged. A negative value indicates that the packet storage does not wrap. That is, when the storage area is full, the collection process stops, as in STOPCOLL (see “PKTS” on page 8 for more information). The minimum integer value is 16M, and the maximum integer value is 2047M. The default value is 16M.

Note: This statement applies only if collection of packet trace data is started automatically during NetView initialization using the INIT.PKTS statement. To specify storage allocation when starting data collection manually, use the STORAGE key word on the PKTS START command. (See “PKTS” on page 8 for more information.)

Commands

This section describes the formats of NetView commands and command lists. You can enter these commands from the command facility or from any other NetView component.

To receive online help for a specific NetView component, enter the following command:

HELP comp

The comp variable is the name of the NetView component.

For online information on a specific NetView command, enter the following command:

HELP cmd

The cmd variable is the name of the command.
**FMTPACKT (NCCF; CNMEFPTC)**

**Syntax**

```
FMTPACKT
   FMTPACKT PKTS_QUERY_command_parameters
   FMTPACKT_pipe_stage_options
```

**Purpose of Command**

The FMTPACKT command collects a subset of the packet trace entries (based on the PKTS QUERY command parameters passed, as described under the PKTS command), converts these trace entries into readable form, and generates reports (based on the FMTPACKT pipe stage options passed) that are returned as NetView messages.

**Operand Descriptions**

**PKTS_QUERY_command_parameters**

These are the same parameters as specified on the PKTS QUERY command. Use of the TRUNCATE parameter will force the NOSEGMENT option on the FMTPACKT pipe stage for the first packet that is actually truncated and all subsequent packets.

**FMTPACKT_pipe_stage_options**

These are the same options as specified on the FMTPACKT pipe stage.

**Usage Notes**

Because of the potential for large amounts (megabytes) of data being processed by this command, the NetView address space size must be large enough or dynamic enough to handle sudden increases and decreases in storage usage. For this same reason, performance of the NetView task running this command might be affected if the SLOWSTG value of the task is not set to 0 or is not set to a very high value. The MAXSTG value of the task should also be set very high or to 0. Because this command can also require a large amount of processor time, the MAXCPU value for this task needs to be set appropriately to balance other work and yet still get timely results returned from this command.

**Examples**

**Example: Displaying Formatted Packets for an IP Address:** To format the packets for IP address 23.128.34.100 for display in a window, enter:

```
WINDOW FMTPACKT RADDR=23.128.34.100 FULL LINESIZE=132
```
LIST (NCCF)

Syntax

NCCF LIST

Purpose of Command
The PKTS keyword has been added to the existing LIST command.

Operand Descriptions

PKTS
Lists each TCP/IP stack name and associated autotask name defined by the PKTS DEFINE command or in member CNMSTYLE as a FUNCTION.AUTOTASK.PKTS statement. The format of each data line is as follows:

PKTS TCPNAME: tname TASKID: tname STATUS: ACTIVE|INACTIVE|DORMANT GTF: NO

The variables in the data line are as follows:

PKTS TCPNAME: tname
The name of the TCP/IP stack.

TASKID: tname
The name of the autotask responsible for collecting packet data for the stack, as defined by the PKTS DEFINE command or a FUNCTION.AUTOTASK.PKTS statement in CNMSTYLE.

STATUS: ACTIVE|INACTIVE|DORMANT
The status of the connection. The possible values are as follows.

ACTIVE
The data collection is currently running.

INACTIVE
The data collection is not currently running, and no packet trace data is available for queries or purges.

DORMANT
The data collection is not currently running, but any previously collected data is still available for queries or purges. A connection can become dormant as a result of the PKTS STOPCOLL command or because of an error.

GTF: NO
The GTF tracing status of the connection. Currently, GTF tracing is not supported for packet data.
The PKTS command is used to control the collection of TCP/IP packet data and to view the collected data.

**Operand Descriptions**

**COUNT** = YES | NO

Specifies whether the response to PKTS QUERY reflects the total number of connections even when this number exceeds the value specified by the MAXRECS operand.

**DEFINE**

Associates a TCP/IP stack name with the name of an autotask that will collect packet data for the stack. This is the same function performed by the FUNCTION.AUTOTASK.PKTS statement in CNMSTYLE. The PKTS START
command is then required to start the function. Data collection for the specified stack stops if it is redefined to another autotask. The same OPID operand can only be defined to one TCP/IP stack.

A value of NONE undefines the specified TCPNAME operand.

**DUMPDATA**

Supports diagnostic information. Use this keyword when instructed to do so by IBM customer support.

**INTFNAME=intfn**

Specifies the interface name as defined by Communications Server. A single asterisk (*) indicates all interfaces. You can also use an asterisk as a wildcard at the end of the interface name; for example, ABC* matches any interface name beginning with the letters ABC.

**LADDR=locaddr**

Specifies the local IP address (or set of addresses) for a QUERY or PURGE command. This address can be expressed in any of several possible formats:

- An IPv4 address in dotted-decimal format: ddd.ddd.ddd.ddd. Each ddd can be any of the following:
  - A decimal number from 0 to 255
  - A hyphen-separated range (such as 240-255)
  - An asterisk (*), representing the range 0–255

Leading zeros can be omitted. If the last ddd is an asterisk, and fewer than four ddd values are specified, the range 0–255 is assumed for each remaining ddd.

- An IPv6 address in dotted-decimal format. The same rules apply as with IPv4 addresses, except with sixteen ddd values instead of four.

- An IPv6 address in hexadecimal format: hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh, where each hhhh is a 0-4 digit hexadecimal number, or a hexadecimal range separated by a hyphen, such as FF00-FFFF. Consecutive groups of zeros can be replaced with a double colon (:). A double colon can be used to signify leading, trailing or embedded groups of zeros, and can only be specified once in an address. A single asterisk (*) can be used in place of hhhh to denote 0-FFFF. If the last hhhh is an asterisk and less than 8 hhhhs are specified, 0-FFFF is assumed for each remaining hhhh. If both an asterisk and a double colon are used in an address, the asterisk will only represent a single hhhh group regardless of its position.

- A single asterisk (*), representing all local IP addresses.

- A TCP/IP symbolic host name. The NetView program will attempt to translate to an IP address using the TCPNAME operand, if the operand is specified without a wildcard. Otherwise, it uses the value for TCPNAME in CNMSTYLE. If a host name resolves to multiple IP addresses, the QUERY or PURGE command will be attempted for all of these addresses.

**LPORT=locport**

Specifies the local port number. locport can be either a decimal number or a single asterisk (*), representing all ports.

**MAXRECS=maxr**

Specifies the maximum number of packet records (data lines) to return from PKTS QUERY. maxr is a number between -9999999 and 9999999 (do not insert commas or periods). Connections are always listed in chronological order. A positive value specifies the set of records beginning with the oldest matching
connection; a negative value specifies the set of records ending with the most recent matching connection. The default value is 100.

**OPID=operid**
Specifies the name of the autotask that will collect packet information for the associated TCP/IP stack.

**PURGE**
Purges packet data records matching the input criteria. If the purge is successful, the BNH774I message is returned.

**QUERY**
Queries packet data records matching the input criteria. If the QUERY is successful, the BNH773I message is returned.

**RADDR=remaddr**
Specifies the remote IP address (or set of addresses) for a QUERY or PURGE. See the description of the LADDR keyword for information on how to specify an IP address.

**RPORT=remport**
Specifies the remote port number. *remport* can be either a decimal number or a single asterisk (*), representing all ports.

**START**
Starts a long-running process to collect packet data from the Communications Server for the specified TCP/IP stack. The data-collection process persists as an autotask defined by a previous PKTS DEFINE command or a FUNCTION.AUTOTASK.PKTS statement in CNMSTYLE.

**STOP**
Stops the collection of packet data for the specified stack. The PKTS STOP command is used to end the data collection started by PKTS START command. After the PKTS STOP command is issued, the QUERY and PURGE commands are no longer valid for the stack.

**STOPCOLL**
Stops the collection of packet data for the specified stack while keeping the data already collected available. Use PKTS STOPCOLL instead of PKTS STOP if you still want to be able to run PKTS QUERY or PKTS PURGE against the stack.

**STORAGE=storg**
The number of megabytes of data space storage to allocate. This value must be a positive or negative integer, followed by the letter M. A positive value indicates that the packet storage wraps. That is, when the storage area is full, arriving packets cause the oldest packets to be purged. A negative value indicates that the packet storage does not wrap. That is, when the storage area is full, the collection process stops, as in the STOPCOLL command. The minimum integer value is 16M, and the maximum integer value is 2047M. The default is value 50M.

**TCPNAME=tname**
Specifies the TCP/IP stack name associated with this request.

For requests, other than DEFINE and QUERY, you can use an asterisk (*) as a wildcard character at the end of the name or as the only specified character, which indicates all stacks specified by a previous PKTS DEFINE command or in your style member as a FUNCTION.PKTS statement. Wildcards are not
supported for DEFINE or QUERY. A wildcard value is supported for QUERY if exactly one stack is defined. Otherwise you must issue separate QUERY commands for each stack.

**TIME=trange**

Specifies the range of times for packets to be included in a QUERY or PURGE command. *trange* consists of two values separated by a comma and enclosed in parentheses; the first value specifies the beginning date and time for the range, and the second value specifies the ending date and time for the range. The following rules apply to these values:

- To specify a date and time, type the date followed by the time, separated by a blank space. The formats of the date and time are controlled by the DEFAULTS and OVERRIDE commands.
- If either value consists only of a date, the current time for that date is used.
- If either value consists only of a time, the assumed date depends upon the specified time. If the time is later than the current time, yesterday’s date is assumed; if the time is earlier than the current time, today’s date is assumed.
- An asterisk before the comma will include all packets from the beginning of data collection. An asterisk after the comma will include all packets up to the present.
- Either value can also be specified as a 16-character hexadecimal store-clock value. This can be used to continue a previous query by specifying a previously returned store-clock value from the BNH773I message.

The following examples would be valid values for *trange*:

- (04/05/04 10:00:00,04/06/04 17:00:00)
- (*,04/05/04)
- (B7ACDD41D4F00A01,*)

**TRUNCATE=qtrunc**

The maximum number of bytes output for each packet in this QUERY response. This includes the CTE and CS headers (see message BNH7731). The smallest allowable *qtrunc* value is 144 (the length of the above headers plus an IPv6 header). The largest value is 65535, which means truncation is not to occur. This is the default value.

**Note:** If you intend to format the resulting packets (see HELP FMTPACKT) the default truncation value is recommended.

**Restrictions**

Collection of IP packet trace data requires a level of Communications Server that supports the NETMONITOR PKTTRCSERVICE profile statement. It also requires that all appropriate Communications Server definitions are in place. Without this support enabled, the PKTS command returns an error message.

**Return Codes**

<table>
<thead>
<tr>
<th>Return Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful processing.</td>
</tr>
<tr>
<td>4</td>
<td>The command did not complete successfully.</td>
</tr>
<tr>
<td></td>
<td>Check the accompanying messages for more information.</td>
</tr>
</tbody>
</table>
Pipeline stages and syntax

This section describes the syntax, keywords, and parameters of the PIPE FMTPACKT stage.

You can obtain information about the PIPE command and its stages by using the NetView online help. To display online help for the PIPE command, enter the following command:

HELP PIPE SYNTAX

To display online help for a specific PIPE stage, enter the following command:

HELP PIPE stage_name

Where:

stage_name

Is any NetView PIPE stage name, for example FMTPACKT.
PIPE FMTPACKT

Syntax

FMTPACKT:

- FULL
- SHORT
- SUMMARY
- TALLY

PORTSEL
- ASCII
- BOTH
- EBCDIC
- HEX
- PORTSEL

LOCAL
- GMT
- LOCAL

SUMMARY

BASIC

DETAIL

CLEANUP

DUMP

FORMAT

NOREASSEM

REASSEM

LINESIZE

SESSION

STATS

SEGMENT

NOSEGMENT

nnnnnnn

(65535, SUMMARY)

= 500

= nnnnnn

= nnn

= nnnn

= nnnn

= nnnn

= nnnn

= nnnn

= 500

= nn

= 65535

= (65535, SUMMARY)

= SUMMARY

= DETAIL

= SUMMARY

= SUMMARY

= DETAIL

= DETAIL

= SUMMARY

= SUMMARY

= DETAIL

= DETAIL

= SUMMARY

= SUMMARY

= DETAIL

= DETAIL

= SUMMARY

= SUMMARY

= DETAIL

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY

= DETAIL

= SUMMARY
IBM-Defined Synonyms

<table>
<thead>
<tr>
<th>Stage Operand</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT</td>
<td>FMT</td>
</tr>
</tbody>
</table>

Purpose of Command
The FMTPACKT stage takes raw TCPIP packet data, converts it into readable form, and generates reports that are passed to the primary output stream. The input stream is discarded unless there is a conversion error and there is a secondary output stream.

If a secondary output stream is connected, input stream data that cannot be converted will be sent to the secondary output stream as the second message in an MLWTO message. The first message is DWO050E, which contains the return code and the reason code from the EZBCTAPI macro. Contact IBM customer support with this information. If there is no secondary output stream, only the DWO050E message is written to the log.

Streams

<table>
<thead>
<tr>
<th>Stream Type</th>
<th>Number Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>1</td>
</tr>
<tr>
<td>output</td>
<td>2</td>
</tr>
</tbody>
</table>

Termination Conditions
FMTPACKT ends when the primary input stream or the primary output stream disconnects.

Operand Descriptions

ASCII
Dumped packet trace data is shown in hexadecimal and interpreted in ASCII translation only.

BASIC
 Specifies the formatting option for packet trace data.

DETAIL
For specific packet types, format each element of the packet data. This applies to DNS, RIP, and SNMP packet data. This is the default value.
SUMMARY
For specific packet types, provide summary data for the packets. This applies to DNS, RIP, and SNMP packet data.

BOTH
Dumped packet trace data is shown in hexadecimal format and interpreted with both ASCII and EBCDIC translations.

CLEANUP=nnnnnn
Defines a record interval. After the specified interval has elapsed, saved packet information in storage is released. The minimum value is 100 records; the maximum value is 1 000 000 records; the default is 500 records. If the record interval is set to 0, cleanup will not occur.

DUMP=nnnnn
Dump the selected packet in hexadecimal format with EBCDIC and ASCII translations, if these were selected or defaulted (the default value of PORTSEL would produce both translations). The IP and protocol headers are dumped separately from the packet data. The value nnnnnn represents the maximum amount of packet data that is to be dumped from each packet. The default value is 65 535 bytes. The minimum value is 0. The maximum value is 65 535. The IP and protocol headers are not subject to this maximum.

The PORTSEL, BOTH, ASCII, EBCDIC, and HEX keywords describe how the dumped packets are translated. The default value is PORTSEL. The display can be changed using these keywords. The default ASCII translation table is used. This table might not match the table being used by the application.

If the STREAMS report is chosen, then the dump of the packets is deferred until the stream of data has been collected.

EBCDIC
Dumped packet trace data is shown in hexadecimal format and interpreted in EBCDIC translation only.

FORMATT
Specifies the format option.

DETAIL
Formats the IP header, protocol header, and the protocol data. This is the default value.

SUMMARY
Formats the IP header and the protocol header.

FULL
Equivalent to DUMP and FORMAT. SUMMARY is the default value.

GMT
The time stamps are converted to GMT time. LOCAL is the default value.

HEX
Dumped packet trace data is shown in hexadecimal format only with no translation.

LINESIZE
Specifies the line width at which the generated reports and data lines are wrapped.

If the output is directed to the operator screen, and a value for LINESIZE is specified that is greater than the width of the NetView operator screen, the
displayed lines appear truncated. This appearance of truncation can be avoided by issuing the command using the WINDOW command which allows scrolling left and right.

* An asterisk indicates that the NetView operator screen width should be used if running under an OST with a real or a virtual screen. Otherwise, LINESIZE will default to 80. A null value for LINESIZE is the same as LINESIZE=*.

**nnn**
A value from 60-250. The default value is 80.

**LOCAL**
The time stamps are converted to local time. This is the default value.

**NOREASSM**
Do not reassemble fragmented IP packets into a complete packet. REASSEM is the default value.

**NOSEGMENT**
Packet trace records that span multiple NetView IP trace records are not recombined. Only the first segment of a packed is used. The rest of the segment records are discarded. SEGMENT is the default value.

**PORTSEL**
For some well known ports, dumped packet trace data is shown in hexadecimal format and interpreted with either ASCII or EBCDIC translations, depending on how the port is defined. If a dump format selection cannot be made, both ASCII and EBCDIC translations are provided. This is the default value.

**REASSEM**
Reassembles IP fragments into a complete packet.

**(nnnnn,DETAIL)**
DETAIL generates the reassembly statistics for each packet when a packet completes reassembly.

**nnnnn** specifies the maximum size allowed for a reassembled packet. This value can be from 576 bytes-65 535 bytes. The default value is 65 535 bytes.

**(nnnnn,SUMMARY)**
SUMMARY generates the reassembly statistics and information for packets that did not complete reassembly. This is the default value.

**nnnnn** specifies the maximum size allowed for a reassembled packet. This value can be from 576 bytes-65 535 bytes. The default value is 65 535 bytes. The default value is 65 535 bytes.

**SEGMENT**
Packet trace records that span multiple NetView IP trace records are recombined. Data from segmented records is saved until all the NetView IP trace records have been read to recreate the original packet. This is the default value.

If the packet trace records as received from the PKTS QUERY command were truncated, the NOSEGMENT option is automatically used.

**SESSION**
List TCP and UDP session information.

**DETAIL**
List each of the packets for TCP and UDP sessions, as well as the summary statistics. This is the default value.
STATE
List the beginning and ending state for each TCP and UDP session.

SUMMARY
Show only the summary statistics for each TCP and UDP session.

SHORT
Equivalent to FORMAT=SUMMARY. SUMMARY is the default value.

STATS
After all of the records have been processed, generates statistical reports.

DETAIL
Lists the number of records selected by record type, device type, job name, link name, protocol number, IP address, and port numbers.

SUMMARY
Lists the IP address and port number pairs with the number of records, the first and last record numbers, and the first and last record times. This is the default value.

STREAMS
Collects the packet data for dumping or formatting after all the trace data has been processed.

(nn,DETAIL)

nn represents the maximum amount of storage used to capture each stream. This value is specified in 1024 (1K) units. The range is 16 KB-512 KB. The default value is 128 KB.

DETAIL generates messages about the status of each stream.

Note: The DUMP option is required to dump the packet data.

(nn,SUMMARY)

nn represents the maximum amount of storage used to capture each stream. This value is specified in 1024 (1K) units. The range is 16 KB-512 KB. The default value is 128 KB.

SUMMARY generates messages about each packet in the streams. This is the default value.

Note: The DUMP option is required to dump the packet data.

SUMMARY
Formats a single line for each trace record. This is the default value.

TALLY
Equivalent to the STATS=DETAIL option. SUMMARY is the default value.

Restrictions
FMTPACKT cannot be the first stage.

If input stream packets are truncated, the packet formatter might generate error messages.
Messages

This section describes NetView messages.

For online help on messages, enter the command that follows.
HELP msgid

where msgid is the identifier of the NetView message for which a help panel is to be displayed.

---

**AAU203I**  module locid DUPLICATE keyword DEFINITION ENCONTERED: label

**Explanation:** The label or identifier on this type of statement must be unique.

**Message Variables:**

- `module` The control section (CSECT) name where an error condition was detected.
- `locid` The location within the CSECT where an error condition was detected.
- `keyword` The related keyword.
- `label` The label that was duplicated.

**System Action:** If this message is issued during session monitor initialization, the NetView program loads the keep class or performance class table but ignores the PCLASS or KCLASS statement with a duplicate label. If this message is issued as the result of a RELOAD command, message AAU234I is issued and the keep class or performance class table is not reloaded (the currently active table is not replaced). Otherwise the duplicate definition is ignored.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** Ensure that each statement has a unique label. Refer to the [IBM Tivoli NetView for z/OS Administration Reference](https://www.ibm.com/support/knowledgecenter/SSLTBW_AU2340) for more information.

---

**AAU204I**  module locid keyword member HAS NOT BEEN DEFINED

**Explanation:** When this message is issued in response to a PKTS command, no TCP/IP stacks have been defined. Refer to the online help for more information about PKTS DEFINE. Otherwise, a MAPSESS statement referred to a performance or keep class that is not defined; that is, no PCLASS or KCLASS statement was encountered.

**Message Variables:**

- `module` The control section (CSECT) name where an error condition was detected.
- `locid` The location within the CSECT where an error condition was detected.
- `keyword` The related keyword.
- `member` The name of the item that was not defined.

**System Action:** If this message is issued during session monitor initialization, the NetView program loads the keep class or performance class table and assigns default values for the undefined PCLASS or KCLASS statement. If this message is issued as the result of a RELOAD command, message AAU234I is issued and the keep class or performance class table is not reloaded (the currently active table is not replaced). Otherwise the command or definition is ignored.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** For session monitor, ensure the performance or keep class referred to is defined with a PCLASS or KCLASS statement. Confirm that this statement precedes any MAPSESS statements that refer to the given performance or keep class. Refer to the [IBM Tivoli NetView for z/OS Administration Reference](https://www.ibm.com/support/knowledgecenter/SSLTBW_AU2340) and otherwise issue one or more PKTS DEFINE commands to define the needed stacks.
BNH148E  Unix Service error for 'service', return code = 'retcode'X, reason code = 'rescode'X.

Explanation: A failure has occurred in an Open Edition service call.

Message Variables:

  service  The UNIX® service name that failed
  retcode  The hexadecimal value of the return code from the service
  rescode  The hexadecimal value of the reason code from the service

System Action: The request fails.

Operator Response: Notify the system programmer.

System Programmer Response: Refer to the appropriate manual in the Open Edition MVS™ library for more information regarding the error.

BNH202E  UNIX SYSTEM SERVICE service FAILED. SOCKET CLOSED.

Explanation: A failure has occurred in a Unix System service call. This message follows message BNH148E.

Message Variables:

  service  The UNIX service name that failed.

System Action: The request fails.

Operator Response: Notify the system programmer.

System Programmer Response: Refer to the appropriate manual in the Open Edition MVS library for more information regarding the error.

BNH548I  'function' IS NOT ACTIVE FOR 'object'

Explanation: The request was not performed for the object.

For the LOGTSTAT command, SMF recording was not active for the specified task.

Message Variables:

  function  The name of the service that was not performed, such as SMF for the LOGTSTAT command.
  object  The name of the object affected.

Operator Response: Determine whether you want the function performed for any of the listed objects.

For the LOGTSTAT command, to activate SMF logging for the specified task, enter one of the following commands:

DEFAULTS LOGTSTAT=YES

or

OVERRIDE TASK=object,LOGTSTAT=YES

Note: It is possible that SMF recording is not active because of MVS user exits that are in effect or MVS commands that were issued. Ask your system programmer to determine the SMF settings for the MVS program.

For the PKTS command, the specified stack is not defined. For more information, refer to the command help for the PKTS DEFINE command.

System Programmer Response: For the LOGTSTAT command, adjust the SMF settings as needed to activate SMF recording.

BNH773I  NUMBER OF PACKETS: nump, MISSED BUFFERS: missbuf, TCPNAME: tname

Explanation: This is a multiline message written in response to the PKTS QUERY command. It consists of one header line followed by a variable number of data lines, each describing a traced packet. The number of packets returned is determined by the MAXRECS operand on the PKTS QUERY command.
Note: PKTS QUERY will display all of the available data, but some error conditions can cause some data to be missing. See any preceding error messages for possible explanations for missing data.

Each data line includes the following information in binary form:

<table>
<thead>
<tr>
<th>Columns</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1–104</td>
<td>The CTE and CS packet headers.</td>
</tr>
<tr>
<td>105–...</td>
<td>The packet data, starting with the IP header.</td>
</tr>
</tbody>
</table>

The binary data can be converted into a printable form using the PIPE EDIT command or the FMTPACKT pipe stage.

**Message Variables:**

- **nump** The number of matching traced data packets. If COUNT=YES was specified on the PKTS QUERY command, this is the number of matching packets. If COUNT=YES was not specified, the value of **nump** is the number of matching packets if less than the value of the MAXRECS keyword, and N/A if not.

- **missbuf** The number of missing data buffers. An increased value of **missbuf** indicates reduced accuracy of the QUERY results.

- **tname** The TCP name specified on the command. If no TCP name is specified, the default TCP name is used.

**System Action:** The PKTS QUERY is complete.

---

**BNH774I**   PURGE COMPLETED. **nump** RECORD(S) OF TYPE **ptype** WERE PURGED FOR TCPNAME **tname**.

**Explanation:** This message is returned in response to the PKTS PURGE command and indicates that the specified number of records have been purged from storage.

**Message Variables:**

- **nump** The number of records that have been purged.
- **ptype** PACKET (packet trace records)
- **tname** The TCP/IP name of the stack.

---

**CNM461I**   CONFLICTING KEYWORDS SPECIFIED IN **cmd** COMMAND

**Explanation:** Mutually exclusive keywords have been entered in the indicated command.

**Message Variables:**

- **cmd** The name of the command specified.

**System Action:** The command is not run.

**Operator Response:** Notify the system programmer if the error occurred in a command list. Otherwise, resolve the keyword conflict and enter the command again.

**System Programmer Response:** Edit the command list and resolve the keyword conflict.

---

**DSI012W**   ’**action1**’ rejected. Task **taskid** is engaged in ’**action2**’

**Explanation:** The task named cannot perform the function named as ’**action1**’ because all its resources are needed for ’**action2**’.

**Message Variables:**

- **action1** The function that cannot be performed
- **taskid** The operator or task where the functions are or would be performed.
- **action2** The function that the task is already doing.

**System Action:** The NetView program ignores the command.

**Operator Response:** Define a new task for the action1 function.
**DSI047E**  
*operation failed: tower not enabled.*

**Explanation:** The requested *operation* failed because the required supporting *tower* was not enabled.

**Message Variables:**
- *operation*  The requested command, function, operation, or service.
- *tower*  The name of the tower or fully qualified subtower that is not enabled.

**System Action:** The *operation* ended.

**Operator Response:** Notify the system programmer.

**System Programmer Response:** Enable the required *tower* in CNMSTYLE or one of its included members.

---

**DSI416I**  
**PROCESSING FAILED FOR 'command' COMMAND**

**Explanation:** The command processing did not complete successfully.

If the command is PKTS, the problem can be that the Communications Server is not at a high enough level, or is not configured to support these functions.

If the command is ASSIGN, AUTOTBL, AUTOTEST, or CPTBL, the previous automation or code point table still resides in storage. Associated messages might indicate the reason for the failure. However, if the command is ASSIGN, AUTOTBL, AUTOTEST, or CPTBL and the table specified in *command* contains no entries, this is the only message you receive.

**Message Variables:**
- *command*  The command you entered

For AUTOTBL, message DSI410I or DSI411I follows to show the current status. For AUTOTEST, message BNH336I or BNH337I follows to show the current status.

**Operator Response:** Notify the system programmer of the problem.

For the AUTOTBL command, if the current automation status is unacceptable, reenter the command with another file name or the OFF operand.

For the AUTOTEST command, use the STATUS operand to find out the status of automation table testing or recording.

**System Programmer Response:** Correct the problem indicated by the associated messages. If this message occurs by itself, it can indicate an empty or missing file.

---

**DSI435I**  
*command COMMAND ENDED DUE TO stringinsert*

**Explanation:** The command ended because of the reason given in the string insert.

**Message Variables:**
- *command*  The name of the failing command.
- *stringinsert*  The reason for the failure.

**System Action:** Processing stops.

**Operator Response:** If another *command* is in progress, wait until it ends and enter the command again. If there is not another *command* in progress, enter the command again. If the second attempt fails, notify the system programmer.

**System Programmer Response:** If this is a change configuration command, the fields can still be updated and sent back to the modem. Collect the problem determination data and contact IBM customer support.

---

**DSI500I**  
*object restarted.*

**Explanation:** The entity identified as *object* was started. This message also indicates that the entity identified as *object* has previously been active.

**Message Variables:**
object  The entity that was started.

System Action:  Processing continues.

Operator Response:  The entity identified as object can now be used.

System Programmer Response:  None

DSI650I  INVALID SYNTAX: 'operand'

Explanation:  You entered a command incorrectly or coded a definition statement incorrectly.

If the message DSI650I is issued during NetView initialization, a reconstituted statement either follows the message or is contained in the message. This reconstituted definition, although it might not be a true representation of the original definition, contains sufficient information to find the original definition in error.

Message Variables:

operand  The operand in which the error was found. Other error responses can be:
- UNBALANCED QUOTES
- PARENTHESES INCOMPLETE
- INVALID DELIMITER
- DELIMITER MISSING
- INCORRECT CONSECUTIVE DELIMITERS
- A quote in position 1 is required.
- An operand without a closing quote.
- A quote is not correctly placed.
- A quote is not followed by a character.
- A quote is not correctly closed.
- A quote is not followed by a character.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
- A quote is not correctly closed.
- A quote is not correctly opened.
Contacting Customer Support

If you have a problem with any Tivoli product, refer to the following IBM® Software Support Web site:


If you want to contact software support, see the IBM Software Support Guide at the following Web site:

http://techsupport.services.ibm.com/guides/handbook.html

The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

• Registration and eligibility
• Telephone numbers and e-mail addresses, depending on the country in which you are located
• Information you must have before contacting IBM Software Support
Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user’s responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX  78758  U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to
IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.

### Trademarks

IBM, the IBM logo, Tivoli, the Tivoli logo, MVS, NetView, and z/OS are trademarks or registered trademarks of International Business Machines Corporation or Tivoli Systems Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.
File Number: S370/4300/30XX-50
Program Number: 5697-ENV

Printed in USA