Version 5.1 OA08904 Readme

• Connection Management for TCP
• NLDM Purge Exception Lists
• Convert VTAM Commands to Upper Case

Version 5 Release 1
Version 5.1 OA08904 Readme

- Connection Management for TCP
- NLDM Purge Exception Lists
- Convert VTAM Commands to Upper Case

Version 5 Release 1
# Contents

**What This Readme Describes** .......................................................... v
- New Enhancements Provided ......................................................... v
- Prerequisites ............................................................................. vi
  - z/OS Prerequisites .................................................................. vi
  - NetView Prerequisites .............................................................. vi
- Backward compatibility ................................................................. vii
- Recommended Publications ......................................................... vii

**Connection Management for TCP** ................................................. 1
- VSAM Cluster Allocation ................................................................. 1
- Customize CNMSTYLE .................................................................. 2
  - Enabling Real-time TCP/IP Connection Management ................... 2
  - CNMSTYLE statements .............................................................. 3
    - TCPCONN.DASD .................................................................. 3
    - TCPCONN.DSRBO ................................................................ 4
    - TCPCONN.ERMSG.THRotte ............................................... 4
    - TCPCONN.GTF .................................................................. 5
    - TCPCONN.KEEP .................................................................. 6
    - TCPCONN.MACRF ................................................................ 7
    - TCPCONN.PDDNM ................................................................ 7
    - TCPCONN.SDDNM ................................................................ 7
- Samples ...................................................................................... 8

**NLDM Purge Exception Lists** ....................................................... 9
- Customize CNMSTYLE .................................................................. 9
  - Examples ................................................................................ 9

**Commands** ................................................................................. 11
- DBAUTO (NCCF; CNME2008) ......................................................... 12
- DBINIT (NCCF; CNME2009) .......................................................... 14
- DBMAINT .................................................................................. 18
- LIST (NCCF) ........................................................................... 20
- PURGE (NLDM). ....................................................................... 22
- PURGEDB (NLDM, NPDA; CNME2007) ...................................... 23
- TCPCONN ................................................................................ 25

**Messages** ................................................................................ 31

**Contacting Customer Support** .................................................... 35

**Notices** .................................................................................... 37
- Trademarks ................................................................................ 39
What This Readme Describes

This document provides important information about enhancements to Tivoli® NetView® for z/OS® Version 5 Release 1. This document contains the most current information for the product and takes precedence over all other documentation.

Please review these notes thoroughly before installing the code changes provided or using the added functions.

New Enhancements Provided

Three enhancements applicable to NetView for z/OS version 5.1 are provided with APAR OA08904.

These first two enhancements, described in “Connection Management for TCP” on page 1 and “NLDM Purge Exception Lists” on page 9, address two distinct features of NetView for z/OS. Although these two enhancements provide different functions, both affect some of the same commands. The section “Commands” on page 11 describes all of the commands and associated parameters affected. “Messages” on page 31 describes new or changed messages.

The third enhancement requires no changes to specification or modification. It is described in its entirety in the section below entitled Convert VTAM® Commands to Upper Case.

These are the enhancements provided:

Connection Management for TCP
This function provides a means of collecting real-time TCP/IP connection information. The function added by APAR OA08904 augments the packet trace data collection that was added by APAR OA04304 (that APAR enabled IP packet trace management and formatting).

NLDM Purge Exception Lists
This function provides greater flexibility in the NLDM PURGE command, such as the ability to purge all sessions older than a given interval except for SSCP-PU sessions or sessions involving any resource beginning with NCP. This function is implemented by using “purge exception lists” that are set up in CNMSTYLE and then referenced in the NLDM PURGE command and its front-end clists (PURGEDB, DBAUTO, DBMAINT, DBINIT, and DBFULL).

Note: In this document, references to CNMSTYLE are intended to mean CxxSTYLE (where xx is your NV2I value) or any of its included members.

Convert VTAM Commands to Upper Case
One additional feature is provided with this APAR, for which no specification or modification is needed. After installing this APAR, the VTAM commands DISPLAY, VARY, and MODIFY will be converted to upper case before being executed, regardless of the OVERRIDE NETVASIS setting.

© Copyright IBM Corp. 2004
Prerequisites

z/OS Prerequisites

You must be running z/OS with Communications Server V1R5 or higher, or Communications Server V1R4 with APAR II13699. This APAR provides a comprehensive list of Communications Server code PTFs.

NetView Prerequisites

These are the PTFs and their corresponding PDO Service Levels that must be applied to your Version 5.1 NetView before installing this APAR:

<table>
<thead>
<tr>
<th>PTF number</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>UW87233</td>
<td>0214</td>
</tr>
<tr>
<td>UW88529</td>
<td>0220</td>
</tr>
<tr>
<td>UW89415</td>
<td>0224</td>
</tr>
<tr>
<td>UW89416</td>
<td>0224</td>
</tr>
<tr>
<td>UW89417</td>
<td>0305</td>
</tr>
<tr>
<td>UW92582</td>
<td>0231</td>
</tr>
<tr>
<td>UW92672</td>
<td>0234</td>
</tr>
<tr>
<td>UW93571</td>
<td>0237</td>
</tr>
<tr>
<td>UW93572</td>
<td>0237</td>
</tr>
<tr>
<td>UW96439</td>
<td>0304</td>
</tr>
<tr>
<td>UA00276</td>
<td>0311</td>
</tr>
<tr>
<td>UA00893</td>
<td>0308</td>
</tr>
<tr>
<td>UA01398</td>
<td>0311</td>
</tr>
<tr>
<td>UA02570</td>
<td>0321</td>
</tr>
<tr>
<td>UA02683</td>
<td>0321</td>
</tr>
<tr>
<td>UA03721</td>
<td>0327</td>
</tr>
<tr>
<td>UA03833</td>
<td>0332</td>
</tr>
<tr>
<td>UA03841</td>
<td>0331</td>
</tr>
<tr>
<td>UA03849</td>
<td>0332</td>
</tr>
<tr>
<td>UA03881</td>
<td>0327</td>
</tr>
<tr>
<td>UA04123</td>
<td>0333</td>
</tr>
<tr>
<td>UA04891</td>
<td>0342</td>
</tr>
<tr>
<td>UA05348</td>
<td>0345</td>
</tr>
<tr>
<td>UA06014</td>
<td>0342</td>
</tr>
<tr>
<td>UA06683</td>
<td>0345</td>
</tr>
<tr>
<td>UA06684</td>
<td>0345</td>
</tr>
<tr>
<td>UA06686</td>
<td>0345</td>
</tr>
<tr>
<td>UA07331</td>
<td>0350</td>
</tr>
<tr>
<td>UA10305</td>
<td>0424</td>
</tr>
<tr>
<td>UA11043</td>
<td>0424</td>
</tr>
<tr>
<td>UA12560</td>
<td>0431</td>
</tr>
<tr>
<td>UA12806</td>
<td>0433</td>
</tr>
</tbody>
</table>

For English language customers:

<table>
<thead>
<tr>
<th>PTF number</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>UW87234</td>
<td>0214</td>
</tr>
<tr>
<td>UW93718</td>
<td>0305</td>
</tr>
<tr>
<td>UA06685</td>
<td>0345</td>
</tr>
<tr>
<td>UA07332</td>
<td>0350</td>
</tr>
<tr>
<td>UA11044</td>
<td>0424</td>
</tr>
<tr>
<td>UA11306</td>
<td>0434</td>
</tr>
</tbody>
</table>

For Japanese language customers:

<table>
<thead>
<tr>
<th>PTF number</th>
<th>Service Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>UW93719</td>
<td>0305</td>
</tr>
<tr>
<td>UW95223</td>
<td>0305</td>
</tr>
</tbody>
</table>
Backward compatibility

This APAR applies only to NetView for z/OS Version 5 Release 1.

Connection Management for TCP
No compatibility issues.

NLDM Purge Exception Lists
Purge Exception Lists are introduced when you install this APAR. Any NLDM inactive sessions recorded prior to installation of this APAR do not contain a session-type; thus, any purge commands will not respect “session type” purge exception list entries and will purge any such NLDM sessions.

Convert VTAM Commands to Upper Case
No compatibility issues.

Recommended Publications

In addition to the NetView publications referred to in this document, you will also need to refer to these books to fully utilize the enhancements added by this APAR.

z/OS Communications Server IP Configuration Guide (SC31–8775)
Contains guidance material to enable you to configure IP address spaces, servers, and applications for the z/OS Communications Server (z/OS CS)

z/OS Communications Server IP Configuration Reference (SC31–8776)
Contains reference material such as statement syntax, options, keywords, and descriptions for the z/OS Communications Server (z/OS CS)
Connection Management for TCP

When you apply this APAR, NetView will provide a means of collecting real-time connection information using the socket interface. This allows you to collect not only start and stop data, but also associated performance data, such as sends and receives, retransmits, and other data. Connection data can now be collected from the z/OS Communications Server continually, thus enabling a picture of connections over time.

To collect real-time connection information using the socket interface, you must be running the appropriate version of z/OS as described in “Prerequisites” on page vi. In addition, you must enable the corresponding services on the NETMONITOR profile statement in Communications Server:

- To enable collection of TCP/IP connection data, specify TCPCONNService.

For more information, see the documentation accompanying the fix for APAR OA04304 (that APAR enabled IP packet trace management and formatting).

In order to utilize this function, you should review the following sections and perform any necessary actions:

- Make needed VSAM definitions described in “VSAM Cluster Allocation.”
- If you use an automation table other than the supplied DSITBL01 and you automate message BNJ022I for NPDA, consider also automating BNJ022I where the text includes 'VSAM DATASET = DSITCON'. If you use DSITBL01, this change is included in this APAR.
- Make CNMSTYLE modifications described in “Customize CNMSTYLE” on page 2.
- Review new or changed parameters for the DBAUTO, DBINIT, LIST, PURGEDB, and TCPCONN commands described in “Commands” on page 11.
- Review new or changed messages described in “Messages” on page 31.

VSAM Cluster Allocation

Note: The following steps assume that you are using sample names provided by NetView. If you have made any changes affecting these sample names, you must make appropriate adjustments.

A new DST named DSITCONT which manages inactive connection records has been created. You must make these VSAM definitions:

- Data set names NETVIEW.CNM01.DSITCONP and NETVIEW.CNM01.DSITCONS have been added to the CNMSJ009 start procedure. If you use any other start JCL, make appropriate changes.
- Sample member CNMSI101 was changed to include DEF (define) steps for the new data sets DSITCONP and DSITCONS. CNMSI101 is referenced in CNMST004, the allocate sample provided by NetView. If you use other allocate JCL, make appropriate changes to that JCL.
- Sample member CNMSI001 was changed to include DELETE...DSITCONP and DSITCONS steps. If you use other delete JCL, make appropriate changes to that JCL.
• Recommended LSR buffer sizes were added to sample CNMSJM01 and will take effect unless you choose to change these buffer sizes. If you change the buffer sizes, change these values in CNMSJM01 and re-assemble it.

• Update CNMSTPWD if you use VSAM passwords.

Customize CNMSTYLE

Enabling Real-time TCP/IP Connection Management

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

Note: Most of the steps listed here are similar to the PKTS functional keywords added to CNMSTYLE for the fix for APAR OA04304 (that APAR enabled IP packet trace management and formatting). In this document, references to CNMSTYLE are intended to mean CxSTYLE (where xx is your NV2I value) or any of its included members.

• Enable the TCPIPCOLLECT tower in CNMSTYLE (if not already done for the PKTS subtower), along with the appropriate subtowers (TCPCONN for TCP/IP connection management, PKTS for packet trace management). See the IBM Tivoli NetView for z/OS Administration Reference for general information on CNMSTYLE and the use of towers.

• Define autotasks to handle the collection of connection data. To do this automatically during NetView startup, add FUNCTION.AUTOTASK statements to CNMSTYLE:
  - To define an autotask to collect connection data for a stack, add the following statement:
    
    (TCPIPCOLLECT.TCPCONN)function.autotask.TCPCONN.stackname=taskname

    where stackname is the name of the TCP/IP stack, and taskname is the name of the autotask responsible for collecting connection data for the stack. Multiple stacks are supported. Use a unique autotask for each stackname that you wish to define.

    You can also define such autotasks interactively by using the TCPCONN DEFINE command.

• If you want to start data collection automatically during NetView startup, add the following statement to CNMSTYLE:

    (TCPIPCOLLECT.TCPCONN)INIT.TCPCONN = Yes

    Note: Note that TCPCONN is now a supported function in the CNMSTYLE keyword for the INIT statement.

    You can also start data collection interactively using the TCPCONN START command.

• To enable automatic database maintenance for the TCP/IP connection data VSAM database, add a corresponding DBINIT command to CNMSTYLE. See the commented-out statements for auxInitCmd of the sample CNMSTYLE for more information. If you choose, you can instead add the DSITCONP and DSITCONS databases to whichever maintenance scheme you already use for NPDA and NLDI data bases.

• To filter the collection of connection data, add TCPCONN.KEEP and TCPCONN.DASD statements to CNMSTYLE.
CNMSTYLE statements

These are CNMSTYLE statements that either are new or have been changed to support Connection Management for TCP.

NetView initialization definitions are contained in DSIPARM member CNMSTYLE. 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.

TCPCONN.DASD

The TCPCONN.DASD statements allow you to specify which connections will be recorded by NetView. For example, you might want to filter out traffic for a high-volume Web server. The connection is identified in these filters by a "4-tuple" consisting of the local address, the local port number, the remote address, and the remote port number. The parameters allow for an asterisk (*) to be used as a wildcard character and for numerical ranges to enable you to list everything you need to record. Ranges and wildcards in these values are specified in much the same way as in the TCPCONN command (described in "TCPCONN" on page 25), with these exceptions:

- The TCPCONN.DASD statement supports ranges for locport and remport.
- The TCPCONN.DASD statement does not support host names. You must specify IP addresses. However, wildcards and ranges are supported.

Each TCPCONN.DASD statement specifies a "stackname"; this corresponds to the TCPNAME value specified on the TCPCONN command. If you have more than one stackname, the statements for each stack apply only to connections for that stack. After being grouped by stackname, the statements are sorted according to the suffixes that you supply with the suffix parameters. Only the first (sorted) filter statement matching any given connection will be applied. To verify that the filters are sorted in the order that you want, use the LIST TCPCONN command after NetView is active. This will show the filters for each active stack in the order in which they will be searched.

Any connection that does not match a TCPCONN.DASD filter or that matches a TCPCONN.DASD filter with the NOT option will not be recorded after it becomes inactive.

The syntax of the TCPCONN.DASD statement is as follows:

TCPCONN.DASD

\[ TCPCONN.DASD.\text{stackname}.\text{suffix}=\text{filter} \]

\[ \text{NOT} \]

Where:

stackname
The name of a TCP/IP stack as defined by Communications Server.

suffix
If identical suffixes are found for the same stackname, only the last one will be applied. Suffixes are used for sorting DASD statements. After sorting, if a connection matches multiple TCPCONN.DASD statements, only the first applies.
NOT
When NOT is specified, a matching connection is not recorded and is not examined for matching any subsequent TCPCONN.DASD statements.

filter
A filter string specifying which connections the statement applies to. The filter string takes the form of a "4-tuple" specification:
locaddr/locport,remaddr/remport

Where:
locaddr
   Specifies the local IP address of the connection.
locport
   Specifies the local port of the connection.
remaddr
   Specifies the remote IP address of the connection.
remport
   Specifies the remote port of the connection.

Note: TCPCONN.DASD applies only to the collection of data for inactive connections. Use the TCPCONN.KEEP statement to specify which connections you want to keep while they are active.

TCPCONN.DSRBO
The TCPCONN.DSRBO statement specifies the number of DSRBOs to be associated with the DSITCONT task.

The syntax for the TCPCONN.DSRBO statement is:

   TCPCONN.DSRBO=number

Where:
number
   Is a decimal number from 1–999 that specifies the projected number of concurrent user requests for services from this DST. The value represents the number of DSRBs to pre-allocate for processing solicited RUs and VSAM requests. If more requests are received than there are DSRBs available, the requests are queued. The default value is 10. You might reference your existing NLD.M.DSRBO setting.

Usage Notes
• If you change the number, update the STRNO keyword on the BLDVRP macro used to create the LSR pool in CNMSJM01.
• There is no RESTYLE support for TCPCONN. If you need to change this setting without recycling NetView, manually change the contents of common global variable CNMSTYLE.TCPCONN.DSRBO and then recycle task DSITCONT.

TCPCONN.ERRMSG.THROTTLE
The TCPCONN.ERRMSG.THROTTLE variable names a command that TCPCONN will invoke if an error is encountered while writing records to its VSAM database.
At a minimum, the command should write error messages to the log. The command that you name can be used, for example, to attract the attention of one or more operators.

The default command, CNMETACI, is a sample REXX procedure that will reduce the volume of messages that might occur when, for example, frequent VSAM updates are all receiving the same error. See the sample CNMETACI for details.

The syntax for TCPCONN.ERRMSG.THROTTLE is:

\[
\text{TCPCONN.ERRMSG.THROTTLE} = \text{command commandargs}
\]

Where:

**command**

Is a command procedure that TCPCONN will invoke whenever an error is encountered while writing records to its VSAM database. The default command is the sample REXX procedure CNMETACI. For additional information, read the prologue commentary for CNMETACI.

**commandargs**

Is the command argument or arguments for the command that you specify. The default argument, 60, is the default argument for the default command, CNMETACI.

**Usage Notes**

- During TCPCONN start processing, NetView determines whether the command named in TCPCONN.ERRMSG.THROTTLE appears to be a REXX procedure. If so, a LOADCL command is issued for the named procedure in order to improve performance.
- The TCPCONN processor invokes this command without any authorization checking.
- There is no RESTYLE support for TCPCONN. If you need to change the command procedure represented by *command*, edit it in library DSICLD. Then issue a DROPCL command and LOADCL command for your command procedure. You cannot change the name of the command procedure that will be invoked without recycling NetView.

**TCPCONN.GTF**

The TCPCONN.GTF statement specifies whether GTF tracing of incoming records and buffers will occur. The syntax of TCPCONN.GTF is as follows:

\[
\text{TCPCONN.GTF} = \text{NO} \quad \text{YES}
\]

**Note:** This statement applies only if collection of TCP/IP connection data is started automatically during NetView initialization using the INIT.TCPCONN statement. To control GTF tracing when starting data collection manually, use the GTF keyword on the TCPCONN START command. (See the [IBM Tivoli NetView for z/OS Command Reference Volume 1](#) for more information.)
**TCPCONN.KEEP**

The TCPCONN.KEEP statements allow you to specify which connections will be kept by NetView. For example, you might want to filter out traffic for a high-volume Web server. The connection is identified in these filters by a "4-tuple" consisting of the local address, the local port number, the remote address, and the remote port number. The parameters allow for an asterisk (*) to be used as a wildcard character and for numerical ranges to enable you to list everything you need to keep. Ranges and wildcards in these values are specified in much the same way as in the TCPCONN command (described in "TCPCONN" on page 25), with these exceptions:

- The TCPCONN.KEEP statement supports ranges for locport and remport.
- The TCPCONN.KEEP statement does not support host names. You must specify IP addresses. However, wildcards and ranges are supported.

Each TCPCONN.KEEP statement specifies a "stackname"; this corresponds to the TCPNAME value specified on the TCPCONNN command. If you have more than one stackname, the statements for each stack apply only to connections for that stack. After being grouped by stackname, the statements are sorted according to the suffixes that you supply with the suffix parameters. Only the first (sorted) filter statement matching any given connection will be applied. To verify that the filters are sorted in the order that you want, use the LIST TCPCONN command after NetView is active. This will show the filters for each active stack in the order in which they will be searched.

Any connection that does not match a TCPCONN.KEEP filter or that matches a TCPCONN.KEEP filter with the NOT option will be filtered out. Such connections are not kept nor will they be matched to any TCPCONN.DASD filter.

The syntax of the TCPCONN.KEEP statement is as follows:

```
TCPCONN.KEEP

TCPCONN.KEEP.stackname.suffix[NOT]filter
```

**Where:**

- **stackname**
  The name of a TCP/IP stack as defined by Communications Server.

- **suffix**
  If identical suffixes are found for the same stackname, only the last one will be applied. Suffixes are used for sorting DASD statements. After sorting, if a connection matches multiple TCPCONN.DASD statements, only the first applies.

- **NOT**
  When NOT is specified, a matching connection is not kept, not examined for any subsequent KEEP statements, and not compared with TCPCONN.DASD statements.

- **filter**
  A filter string specifying which connections the statement applies to. The filter string takes the form of a "4-tuple" specification:

```
locaddr/locport,remaddr/remport
```
Where:

locaddr
- Specifies the local IP address of the connection.

locport
- Specifies the local port of the connection.

remaddr
- Specifies the remote IP address of the connection.

remport
- Specifies the remote port of the connection.

TCPCONN.MACRF
The TCPCONN.MACRF statement specifies the local shared resource (LSR) options for TCP/IP connection management.

The syntax for the TCPCONN.MACRF statement is:

```
TCPCONN.MACRF=

Where:

LSR
- Enables the reclaiming of data and index buffers by keeping a pool of the most recently referenced records in storage. This is effective in reducing physical I/O. This is the recommended option.

Note: Do not use the DFR option or the NSR option unless instructed by IBM® Service.

TCPCONN.PDDNM
The TCPCONN.PDDNM statement specifies the TCP/IP connection management primary data set.

The syntax for the TCPCONN.PDDNM statement is:

```
TCPCONN.PDDNM=name

Where:

name
- Indicates the 1–8 character DD name of the primary data set to be used by VSAM services. The default DSITCONP is strongly recommended.

Usage Notes
- If necessary, specify the VSAM password in CNMSTPWD.

TCPCONN.SDDNM
The TCPCONN.SDDNM statement specifies the TCP/IP connection management secondary data set.

The syntax for the TCPCONN.SDDNM statement is:
TCPCONN. SDDNM=name

Where:

name
Indicates the 1–8 character DD name of the secondary data set to be used by VSAM services. The default DSITCONS is strongly recommended.

Usage Notes
- If necessary, specify the VSAM password in CNMSTPWD.

Samples

The response to a TCPCONN QUERY is multiline message BNH772I, described in “Messages” on page 31. This message is designed for parsing by REXX and not for reading by a person. A REXX sample, CNMSTCPC, is provided to format the BNH772I information for readability.

This sample basically accepts the same parameters as the TCPCONN QUERY command and outputs a scrollable formatted response which includes column labels and uses colors to indicate active and inactive connections. This output can be easily customized. For more information, see the prologue commentary in sample CNMSTCPC and the help for TCPCONN.
NLDM Purge Exception Lists

NLDM Purge Exception Lists provide greater flexibility in the NLDM PURGE command, such as the ability to purge all sessions older than a given interval except for SSCP-PU sessions or sessions involving any resource beginning with NCP. This function is implemented by using “purge exception lists” that are set up in CNMSTYLE, and then referenced in the NLDM PURGE command and its front-end clists (DBAUTO, DBFULL, DBINIT, DBMAINT, and PURGEDB).

In order to utilize this function, you should review the following sections and perform any necessary actions:

- Make CNMSTYLE modifications described in “Customize CNMSTYLE.”

  Note: In this document, references to CNMSTYLE are intended to mean CxSTYLE (where xx is your NV2I value) or any of its included members.

- Review new or changed parameters for the DBAUTO, DBINIT, DBMAINT, PURGE (NLDM), and PURGEDB commands described in “Commands” on page 11.

- Review new or changed messages described in “Messages” on page 31.

Customize CNMSTYLE

The NLDM PURGE command supports a PEXLSTxx parameter, which specifies that any session that matches any element of purge exception list xx is NOT to be purged. Specify such lists in CNMSTYLE, keeping each list contiguous.

For each list, specify PEXLSTxx.y where two non-blank characters xx specify a list and y is a unique suffix for each element in that list. In the examples, identifiers of 01 and 02 are used. Each element is identified by a unique suffix which consists of 1-8 non-blank characters. In the examples, A and B are used as suffixes. You can use alpha, numeric, or national (@#$) characters for all 2-character PEXLST identifiers and suffixes.

The value for an entry can be either a resource pair or a session type. Resource pairs are specified exactly the same way as they are in the NLDM PURGE command (see command help). Session type is one of the following: SSCP-SSCP, CP-CP, SSCP-PU, SSCP-LU or LU-LU.

Examples

Note: Both of these examples assume this PEXLST specification:

```
NLDM.PEXLST01.A = NCP* *
NLDM.PEXLST01.B = SSCP-SSCP
NLDM.PEXLST02.A = HOST1 NCP*
NLDM.PEXLST02.B = CP-CP
```

In this example, all sessions involving a HOST* resource are purged except for those between HOST1 and an NCP* resource, or any CP-CP sessions.

```
NLDM PURGE SESSION HOST* * PEXLST02 BEFORE *
```

This example will purge all sessions before date1/time1 except for those involving an NCP* resource or any SSCP-SSCP sessions:
## Commands

This section describes NetView commands that are new or changed.

### Commands and parameters changed for the Connection Management for TCP enhancement

**Note:** This document describes only the commands and parameters added or changed. A complete list of all of the commands and command parameters can be found in *IBM Tivoli NetView for z/OS Command Reference Volume 1*.

**DBAUTO (NCCF; CNME2008)**

A new parameter TCPCONN is added. See “DBAUTO (NCCF; CNME2008)” on page 12.

**DBINIT (NCCF; CNME2009)**

Help for this command is new. TCPCONN support is new. See “DBINIT (NCCF; CNME2009)” on page 14.

**LIST**

A new parameter TCPCONN is added. See “LIST (NCCF)” on page 20.

**PURGEDB (NLDM, NPDA; CNMD2007)**

A new parameter TCPCONN is added. See “PURGEDB (NLDM, NPDA; CNME2007)” on page 23.

**TCPCONN**

This command is new. See “TCPCONN” on page 25.

### Commands and parameters changed for the NLDM Purge Exception Lists enhancement

**DBAUTO (NCCF; CNME2008)**

A new parameter PEXLSTxx is added. See “DBAUTO (NCCF; CNME2008)” on page 12.

**DBINIT (NCCF; CNME2009)**

Help for this command is new. PURGExx support is new. See “DBINIT (NCCF; CNME2009)” on page 14.

**DBMAINT**

A new parameter PEXLSTxx is added. See “DBMAINT” on page 18.

**PURGE (NLDM)**

A new parameter PEXLSTxx is added. See “PURGE (NLDM)” on page 22.

**PURGEDB (NLDM, NPDA; CNMD2007)**

A new parameter PEXLSTxx is added. See “PURGEDB (NLDM, NPDA; CNME2007)” on page 23.

For online help on commands, enter

```
HELP cmdid
```

where *cmdid* is the NetView command for which a help panel is to be displayed.
**DBAUTO (NCCF; CNME2008)**

**Syntax**

```
DBAUTO
```

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NLDM</td>
<td>NPDA</td>
<td>EZLSTS</td>
<td>FKVSNU</td>
</tr>
<tr>
<td>TCPCONN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVE</td>
<td>TARA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLEAR</td>
<td>PURGE</td>
<td>REORG</td>
<td>SWITCH</td>
</tr>
<tr>
<td>REORG</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWITC H</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>,password</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>,TRK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>,CVE</td>
<td>,50</td>
<td>,primary_space</td>
<td></td>
</tr>
<tr>
<td></td>
<td>,50</td>
<td>,secondary_space</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>,passthru</td>
</tr>
</tbody>
</table>
```

**Purpose of Command**

The DBAUTO command enables you to purge entries from your VSAM database, reorganize the database, switch to a secondary database, or clear the database altogether.

**Operand Descriptions**

**Note:** All parameters shown in the preceding syntax diagram are fully described in *IBM Tivoli NetView for z/OS Command Reference Volume 1*. This document describes only the new parameter TCPCONN and the PEXLSTxx sub-parameter to the PURGE parameter, both of which are added to the DBAUTO command.

**TCPCONN**

TCP/IP connection management VSAM database.

**PURGE**

Enables you to selectively purge entries in your VSAM database. Using the *days* option, you can automatically purge entries in your VSAM database that are older than a specified number of days. After the database entries are purged, the VSAM database is reorganized (see REORG below).

**Note:** PURGE can be used only by NLDM and NPDA. There is no PURGE option for SAVE or TARA.

*days* The number of days of activity you want to keep in the VSAM database. This parameter is optional and can only be specified with the PURGE function. This value overrides the common global variables: NLDM PURGE_DAYS, NPDA PURGE_DAYS. The defaults are 2 for NLDM and 5 for NPDA.

**PEXLSTxx**

Specifies a purge exception list that is defined in CNMSTYLE.

The *xx* corresponds to the CNMSTYLE exception list definition (NLDM.PEXLSTxx.suffix).

**Note:** PEXLSTxx is valid only with the NLDM keyword.
Restrictions

Note: All restrictions that apply to the DBAUTO command are listed in IBM Tivoli NetView for z/OS Command Reference Volume 1. No additional restrictions to those listed in that book are imposed.

Examples

Note: This example demonstrates a function that has been added by this APAR. Other examples pertaining to the DBAUTO command are listed in IBM Tivoli NetView for z/OS Command Reference Volume 1.

Example: Purging Session Monitor Entries Using an Exception List
To purge everything in the session monitor database older than 2 days except for entries with session information between HOST1 and an NCP resource, or any CP-CP session, enter:

DBAUTO NLDM,PURGE,2,PEXLST02

The corresponding CNMSTYLE statements containing the exception lists are:

NLDM.PEXLST02.A = HOST1 NCP*
NLDM.PEXLST02.B = CP-CP
DBINIT (NCCF; CNME2009)

Syntax

```
DBINIT
   NLDM       password       CYL       primary_space       secondary_space
   NPDA
   TARA
   SAVE
   TCP_CONN
```

```
   automation       action       days       timer_start       timer_action       timer_days
```

```
   timer_time_of_day       timer_interval
   passthru
```

Purpose of Command
The DBINIT command enables you to set global variables and perform periodic database maintenance using the EVERY timer and the DBAUTO command. This command is invoked by CNMSTYLE when the NetView program starts.

Operand Descriptions

NLDM
- Specifies the session monitor VSAM database.

NPDA
- Specifies the hardware monitor VSAM database.

TARA
- Specifies the Threshold Analysis and Remote Access (4700 Support Facility) VSAM database.

SAVE
- Specifies the Save/Restore VSAM database.

TCP_CONN
- Specifies the TCP/IP connection management VSAM database.

password
- Specifies your VSAM database password. This password can be in the range of 1–8 alphanumeric characters. Specify NONE if a password is not used.

CYL
- Specifies space allocation-type of cylinder.

TRK
- Specifies space allocation-type of track.

primary_space
- Specifies the amount of primary space allocated to temporarily hold the database contents while REORG is running.
**secondary_space**
Specifies the amount of secondary space allocated to temporarily hold the database contents while REORG is running.

**automation**
Specifies whether to enable the DBFULL function when DBFULL is invoked by message automation from the automation table:
- Y – specifies to enable DBFULL
- N – specifies to disable the DBFULL.

**Note:** DBFULL is invoked when the specified database is full. DBFULL calls DBAUTO to perform database maintenance.

**action**
Specifies the action for DBFULL to perform when invoking the DBAUTO command:
- **CLEAR**
  Erases the contents of the entire VSAM database.
  For NLDM, when the CLEAR request completes, NLDM session recording is started. If NLDM session recording was active prior to the CLEAR request, it is restarted when the CLEAR request completes. If NLDM session recording was not active prior to the CLEAR request, it is started when the CLEAR request completes.
- **PURGE**
  Enables you to purge entries in your VSAM database.
  After the database entries are purged, the VSAM database is reorganized (see REORG below).
- **PURGE<xxx>**
  For the NLDM database, you can specify a purge exception list that is defined in CNMSTYLE. The <xx> corresponds to the CNMSTYLE exception list definition (NLDM.PEXLST<xx>).
- **REORG**
  Enables you to reorganize your VSAM database, which claims free space and allows this space to be reused.
- **SWITCH**
  Enables you to switch from an active to an inactive VSAM database.

**days**
The number of days of activity to keep in the VSAM database for the PURGE or PURGE<xxx> action.
For CLEAR, REORG, and SWITCH, specify 0.

**timer_start**
Specifies whether to initiate a DBAUTO command on an EVERY timer for periodic database maintenance:
- Y – initiate the DBAUTO command
- N – do not initiate the DBAUTO command

**timer_action**
Specifies the action for the timed DBAUTO command:
- **CLEAR**
  Erases the contents of the entire VSAM database.
  For NLDM, when the CLEAR request completes, NLDM session recording is started. If NLDM session recording was active prior to the CLEAR request, it is restarted when the CLEAR request completes. If NLDM session recording was not active prior to the CLEAR request, it is started when the CLEAR request completes.
- **PURGE**
  Enables you to purge entries in your VSAM database.
  After the database entries are purged, the VSAM database is reorganized (see REORG below).
Note: PURGE can only be used with NLDM, NPDA, and TCPCONN.

PURGExx
For the NLDM database, you can specify a purge exception list that is defined in CNMSTYLE. The xx corresponds to the exception list statement (NLDM.PEXLSTxx).

Note: PURGExx can only be used with NLDM.

REORG
Enables you to reorganize your VSAM database, which claims free space and allows this space to be reused.

SWITCH
Enables you to switch from an active to an inactive VSAM database.

timer_days
Specifies the number of days of activity to keep in the VSAM database for the PURGE or the PURGExx timer_action function and the resulting timed DBAUTO command. For CLEAR, REORG, and SWITCH, specify 0.

timer_time_of_day
Specifies the time of day in hours (00-24), minutes (00-59), and seconds (00-59) that DBAUTO will perform database maintenance. The format is controlled by the setting of the time operands of the DEFAULTS and OVERRIDE commands. If you specify 24 for hours, specify 00 for minutes and seconds.

timer_interval
Specifies the number of days between DBAUTO invocations for database maintenance.

passthru
Specifies additional parameters that are passed to the timed DBAUTO command.

Restrictions
The following restrictions apply to the DBINIT command:
• All parameters are positional.
• All parameters are required except passthru.

Examples

Example: Reorganizing a Database
To reorganize the Save/Restore VSAM database, designating 50 cylinders for the primary and secondary databases, enter:

DBINIT SAVE NONE CYL 50 50 N REORG 0 Y REORG 0 03:30:00 1

This process will be repeated at 3:30 every day.

Example: Purging Database Entries
To purge everything in the NLDM database except for the last 2 days of entries, enter:

DBINIT NLDM NONE CYL 50 50 Y PURGE 2 Y PURGE 2 02:00:00 1

This process will be repeated at 2:00 every day.

Example: Purging Session Monitor Database Entries Using an Exception List
To purge everything in the session monitor database before the last two days, except entries with session information between HOST1 and NCP resources or CP-CP sessions, enter:

DBINIT NLDM NONE CYL 50 50 Y PURGEO2 2 Y PURGEO2 2 02:00:00 1

The corresponding CNMSTYLE statements containing the exception lists are:
NLDM.PEXLST02.A = HOST1 NCP*
NLDM.PEXLST02.B = CP-CP
DBMAINT

Syntax

```
DBMAINT NLDM,1,days,45,purge_time,COMPRESS=YES,COMPRESS=NO,PESLSTxx
```

IBM-Defined Synonyms

<table>
<thead>
<tr>
<th>Command or Operand</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBMAINT</td>
<td>CLEARSTS</td>
</tr>
</tbody>
</table>

Purpose of Command

The DBMAINT command performs VSAM database maintenance for the NetView hardware monitor, the NetView session monitor, or the AON status file. To use the DBMAINT command from the operator interface, refer to the


Operand Descriptions

**Note:** All parameters shown in the preceding syntax diagram are fully described in

[IBM Tivoli NetView for z/OS Command Reference Volume 1](#).

This document describes only the new parameter PESLSTxx which was added to the DBMAINT command.

**PESLSTxx**

Specifies a purge exception list that is defined in CNMSTYLE.

The xx corresponds to the CNMSTYLE exception list definition (NLDM.PESLSTxx.suffix).

**Note:** The PESLSTxx parameter can be used only with NLDM.

Examples

**Note:** These examples demonstrate a function added by this APAR. Other examples pertaining to the DBMAINT command are listed in

[IBM Tivoli NetView for z/OS Command Reference Volume 1](#).

**Example: Deleting Session Monitor Records Using an Exception List**

To delete records from the NetView session monitor database (except those involving NCP resources or SSCP-SSCP sessions) that are older than three days, enter:

```
DBMAINT NLDM,3,30,PESLST01
```

The exception lists are located in CNMSTYLE:
NLDMPLEXLST01.A = NCP
NLDMPLEXLST01.B = SSCP-SSCP
LIST (NCCF)

Syntax

NCCF LIST

Note: Subsections for ListOP, Memstat, Security, and Status are not shown in this syntax diagram because they are not changed. These parameters are fully described in IBM Tivoli NetView for z/OS Command Reference Volume 1.
Purpose of Command
The LIST command gives information about your NetView session, including the current primary and backup focal points.

Operand Descriptions

Note: All parameters shown in the preceding syntax diagram are fully described in IBM Tivoli NetView for z/OS Command Reference Volume 1. This document describes only the new parameter TCPCONN that is added to the LIST command.

TCPCONN
Displays information related to the TCP/IP stacks being managed by NetView. For each stack, LIST TCPCONN displays a data line providing details about the stack, followed by the KEEP and DASD filtering lines associated with the stack in CNMSTYLE. The format of the data line is as follows:

TCPCONN TCPNAME: tname TASKID: tname STATUS: ACTIVE|INACTIVE|DORMANT GTF: YES|NO

The variables in the data line are as follows:

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

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

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

ACTIVE
Data collection is currently running.

INACTIVE
Data collection is not currently running, and no TCP/IP connection data is available for queries or purges.

DORMANT
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 an error that causes data collection to stop.

GTF: YES|NO
The GTF tracing status of the connection.
PURGE (NLDM)

Syntax

NLDM PURGE

Purpose of Command

The NLDM PURGE command deletes route data, session data, or both from the session monitor database.

The session monitor purge runs asynchronously. When the network operator receives the PURGE STARTED message indicating that the session monitor purge has started, the operator can continue using the session monitor. When the session monitor purge has completed, the authorized receiver receives a PURGE COMPLETED message. This message indicates the type and count of the data purged. If a network operator issues a session monitor PURGE and a PURGE COMPLETED message has not been received by the authorized receiver for a prior session monitor PURGE command, the network operator still receives a PURGE STARTED message for the PURGE command. However, the new session monitor PURGE does not begin until the prior purge has completed.

Operand Descriptions

Note: All parameters shown in the preceding syntax diagram are fully described in IBM Tivoli NetView for z/OS Command Reference Volume 1. This document describes only the new parameter PEXLSTxx, added to the PURGE (NLDM) command.

PEXLSTxx

Specifies a purge exception list that is defined in CNMSTYLE.

The xx corresponds to the CNMSTYLE exception list definition (NLDM.PEXLSTxx.suffix).

Examples

Note: This example demonstrates a function added by this APAR. Other examples pertaining to the PURGE (NLDM) command are listed in IBM Tivoli NetView for z/OS Command Reference Volume 1.

The format of dates and times specified in this example assumes the default setting for date and time formats on the DEFAULTS and OVERRIDE commands.

Example: Purging Sessions Using an Exception List

To purge all session data except for entries with session information between HOST1 and NCP resources or CP-CP sessions, enter:

PURGE SESSION * * PEXLST02 BEFORE *

The exception lists are located in CNMSTYLE:

NLDM.PEXLST02.A = HOST1 NCP*
NLDM.PEXLST02.B = CP-CP
PURGEDB (NLDM, NPDA; CNME2007)

Syntax

PURGEDB

IBM-Defined Synonyms

<table>
<thead>
<tr>
<th>Command or Operand</th>
<th>Synonym</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUTE</td>
<td>RT</td>
</tr>
<tr>
<td>SESSION</td>
<td>SESS</td>
</tr>
<tr>
<td>EVENT</td>
<td>EV</td>
</tr>
<tr>
<td>STAT</td>
<td>ST</td>
</tr>
<tr>
<td>ATTACHED</td>
<td>ATT</td>
</tr>
<tr>
<td>EV/ST/GMFALERT</td>
<td>EVSTGMF and EV/ST/GMF</td>
</tr>
</tbody>
</table>

Purpose of Command

The PURGEDB command list deletes hardware monitor, session monitor, and TCP/IP connection data from the database.

If you purge events, statistics, or GMFALERT records, they are purged from the hardware monitor database. Specify GMFALERT to purge GMFALERT records from the hardware monitor database. Specify EV/ST/GMFALERT to purge events, statistics, and GMFALERT records from the hardware monitor database.

The session monitor purge, when initiated by the PURGE command, runs asynchronously. When you receive the PURGE STARTED message indicating that the session monitor purge began, you can continue using the session monitor. When the session monitor purge ends, you and an authorized receiver receive a PURGE COMPLETED message. This message indicates the type and count of the data purged.
Operand Descriptions

Note: All parameters shown in the preceding syntax diagram are fully described in [IBM Tivoli NetView for z/OS Command Reference Volume 1]. This document describes only the new parameters PEXLSTxx and TCPCONN added to the PURGEDB command.

PEXLST:xx
 specifies a purge exception list that is defined in CNMSTYLE.

The xx corresponds to the CNMSTYLE exception list definition (NLDM.PEXLSTxx.suffix).

TCPCONN
 indicates that TCP/IP connection data is to be purged.

Restrictions

Note: All restrictions pertaining to PURGEDB can be found in [IBM Tivoli NetView for z/OS Command Reference Volume 1]. There are no new restrictions added to the PURGEDB command.

Examples

Note: These examples demonstrate a function added by this APAR. Other examples pertaining to the PURGEDB command are listed in [IBM Tivoli NetView for z/OS Command Reference Volume 1].

The format of dates and times specified in the following examples assumes the default setting for date and time formats on the DEFAULTS and OVERRIDE commands.

Example: Purging Session Monitor Data Using an Exception List

To purge all session monitor session data except for entries with session information between HOST1 and an NCP resource or any CP-CP session, enter:

```
PURGEDB SESSION * * PEXLST02 BEFORE *
```

The exception lists are located in CNMSTYLE:

```
NLDM.PEXLST02.A = HOST1 NCP*
NLDM.PEXLST02.B = CP-CP
```

Response

All session monitor session data except for sessions between HOST1 and NCP resources or CP-CP sessions are purged.
TCPCONN

Syntax
TCPCONN

Purge:
PURGE
  
  LADDR=*
  LPORT=*
  RADDR=*
  RPORT=*
  LUNAME=*
  APPLNAME=*
  STARTIME=(*,*)
  ENDTIME=(*,*)
  SELECT=ACT
  FORCE=NO
  FORCE=YES

Query:
QUERY
  
  LADDR=*
  LPORT=*
  RADDR=*
  RPORT=*
  LUNAME=*
  APPLNAME=*
  STARTIME=(*,*)
  ENDTIME=(*,*)
  SELECT=ACT
  MAXRECS=−100
  COUNT=NO
  MAXRECS=−100
  COUNT=YES

Purpose of Command
The TCPCONN command is used to control the collection of TCP/IP connection data and to view the collected data.

Operand Descriptions
APPLNAME=applnm
  Specifies the TN3270 application name.
Notes:
1. When this parameter is used with SELECT=ACT, only the default value is valid.
2. When this parameter is used with SELECT=ALL, the value is ignored for active connections. All active connections which match the other criteria are returned.

COUNT=YES|NO
Specifies whether the response to TCPCONN QUERY reflects the total number of connections even when this number exceeds the value specified by MAXRECS.

DEFINE
Associates a TCP/IP 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.TCPCONN statement in CNMSTYLE.) TCPCONN START is then required to start the function. Data collection for the specified stack stops if it becomes defined on another autotask. The same OPID can only be defined to one TCPNAME, and to either TCPCONN or PKTS.

A specification of OPID=NULL undefines the specified TCPNAME.

DUMPDATA
Supports diagnostic information. Use this keyword when instructed to by IBM Service.

ENDTIME=range
Specifies the range of end times for connections to be included in a QUERY or PURGE command. range consists of two values separated by commas 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. 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 included.
- 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 connections from the beginning of data collection. An asterisk after the comma will include all connections up to the present (possibly including active connections, depending on other options).
- 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 message BNH772I, described in "Messages" on page 31. This message is designed for parsing by REXX and not for reading by a person. A REXX sample, CNMSTCPC, is provided to format the BNH772I information for readability.

The following examples would be valid values for range:
- (04/05/04 10:00:00, 04/06/04 17:00:00)
- (*,04/05/04)
- (B7ACDD41D4F00A01,*)

Notes:
1. When this parameter is used with SELECT=ACT, only the default value is valid.
2. When this parameter is used with SELECT=ALL, the value is ignored for active connections. All active connections which match the other criteria are returned.

FILTERS
Enables the TCPCONN.KEEP and TCPCONN.DASD statements in CNMSTYLE for the processes described above which are active processes (or stacks). These filters will also be in effect for all processes (or stacks) which start later.

Note: No TCPNAME values are allowed with Filters except a single asterisk (*).
FORCE=YES|NO
Specifies whether a search that might take a long time to complete is really intended. FORCE=YES is required if all of the following are specified or defaulted and LADDR and RADDR are non-specific, meaning that the value contains * (asterisk) or – (dash) and is not a host name.
- SELECT=INACT or SELECT=ALL
- LPORT=* 
- ENDTIME=(*,*)

GTF=YES | NO
Specifies whether GTF tracing of incoming records and buffers will occur. If collection is already in progress for the specified stack, this keyword (with TCPCONN START) will start or stop GTF tracing without interrupting data collection.

The event identifier used for records written to GTF is X'5F8’.

HASHSIZE
This operand is used to improve performance. This specifies the approximate maximum number of active connections expected at a given time. The value cannot contain any spaces or commas. The default value is 50000.

LADDR=locaddr
Specifies the local IP address (or set of addresses) for a QUERY or PURGE. 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 may 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 IP v4 addresses, except with sixteen ddd values instead of four.
- An IPv6 address in hexadecimal format: hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh, where each hhhh is a 0-4 digit hexadecimal number, or a hexadecimal range separated by a dash, such as FF00-FFFF. Consecutive groups of zeros may be replaced with a double colon (:). A double colon can be used to signify leading, trailing or embedded groups of zeros, and may only be specified once in an address. A single asterisk (*) may be used in place of hhhh to denote 0-FFFF. If the last hhhh is an asterisk and less than 8 hhhh’s 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. NetView will attempt to translate to an IP address using the TCPOPND operand, if the operand is specified without a wildcard. Otherwise, it uses the value in CNMSTYLE for TCPCON. If a host name resolves to multiple IP addresses, the QUERY or PURGE 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.

Note: QUERY and PURGE requests are optimized by local port. Whenever possible, specify a specific port rather than *.

LUNAME=LU unm
Specifies the TN3270 logical unit name.

Notes:
1. When this parameter is used with SELECT=ACT, only the default value is valid.
2. When this parameter is used with SELECT=ALL, the value is ignored for active connections. All active connections which match the other criteria are returned.

**MAXRECS=maxr**

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

**OPID=operid**

Specifies the name of the autotask that will collect connection information for the associated TCP/IP stack.

**PURGE**

Purges connection records matching the input criteria from storage or DASD. If the purge is successful, one or two BNH774I messages are returned (one for active connections and one for inactive connections).

**Note**: TCPCONN PURGE might take a long time to complete when purging inactive records from DASD. To avoid this, it is recommended to run this command on a timed basis as part of routine database maintenance.

**QUERY**

Queries connection records matching the input criteria. If the QUERY is successful, message BNH772I is returned. This message, described in [*Messages* on page 31] is designed for parsing by REXX and not for reading by a person. A REXX sample, CNMST CPC, is provided to format the BNH772I information for readability.

**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.

**SELECT=actinact**

Specifies active connections, inactive connections, or both. The valid values for *actinact* are:

- ACT
- INACT
- ALL

**START**

Starts a long-running process to collect connection start and stop data from the Communications Server for the specified TCP/IP stack. The data-collection process persists as an autotask defined by a previous TCPCONN DEFINE command or a FUNCTION,AUTOTASK.TCPCONN statement in CNMSTYLE.

If the autotask is already running, TCPCONN START can be used to start or stop GTF tracing without interrupting data collection.

The TCPCONN.KEEP and TCPCONN.DASD filter settings which are current in CNMSTYLE will go into effect when TCPCONN.START is run.

**STARTTIME=srange**

Specifies the range of start times for connections to be included in a QUERY or PURGE command. *srange* consists of two values separated by commas 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. For more information and examples, see the description of the ENDTIME keyword.
STOP

Stops the collection of connection data for the specified stack. TCPCONN STOP is used to end the data collection started by TCPCONN START.

TCPNAME=tname

For TCPCONN DEFINE, specifies the TCP/IP name to associate with an autotask that will collect connection data for the stack.

For other requests, specifies the TCP/IP name for the request. tname must be a TCP/IP name specified by a previous TCPCONN DEFINE command or a FUNCTION.AUTOTASK.TCPCONN statement in CNMSTYLE. A single asterisk (*) can be used to specify all defined stacks. You can also use an asterisk as a wildcard at the end of the TCP/IP name; for example, ABC* matches any TCP/IP name beginning with ABC.

Restrictions

The following restrictions apply to the TCPCONN command:

- Collection of TCP/IP connection data requires a level of Communications Server (CS) which supports the NETMONITOR TCPCONNSERVICE profile statement. It also requires that all appropriate Communications Server definitions are in place. (See [IBM Tivoli NetView for z/OS Installation: Configuring Additional Components](#) for more information.) Without this support enabled, TCPCONN returns an error message.

- For any application that generates a high volume of connections (such as a Web server), filtering should be enabled in Communications Server. For example, the MINLIFETIME parameter specifies how long the server waits before reporting a new connection. If a connection closes before this time, it is not reported. The default is 3 seconds. Without CS filtering, TCPCONN START will cause NetView to receive notification of every connection start and stop on the stack, which could cause performance problems. Note that the TCPCONN.KEEP statements in CNMSTYLE provide filtering only on the NetView side.

- Note that there may be a delay between the time a connection becomes inactive (will no longer match a query or purge as an active connection), and the time the connection is recorded to DASD as an inactive connection (and will then match a query or purge as an inactive connection). The length of this delay will depend on the workload of task DSITCONT (the flow of connections becoming inactive which passed both KEEP and DASD filters).

- The following error message in the network log indicates that some of the four keys that make up an inactive connection on DASD are not present. In the QUERY case, the connection is not displayed. In the PURGE case, the remaining keys are purged. The key that was not found follows this message:

  DW0970I TCPCONN : DSIVSMX(2) FAILED WITH RETURN CODE 992

- TCPCONN PURGE might take a long time to complete when purging inactive records from DASD. It is recommended that this command is run on a timed basis as part of routine database maintenance.

- When TCPCONN PURGE encounters errors purging inactive records from DASD, it continues to attempt to purge records if possible, but in some cases terminates. The presence of error messages might cause the count in message BNH774I to be incorrect. In this case, TCPCONN QUERY can be used to determine what was actually purged.

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. Check the accompanying messages for more information.</td>
</tr>
</tbody>
</table>
Messages

This section describes messages that are new or changed.

New message for Connection Management for TCP
- BNH772I

New or changed messages for NLDM Purge Exception Lists
- AAU270I
- AAU927I

For online help on messages, enter
HELP msgid

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

AAU270I SYNTAX ERROR INVOLVING BEFORE PHRASE OR PEXLST PARAMETER

Explanation: The syntax for the NLDM PURGE command is incorrect:
- The BEFORE phrase is not specified or is in the wrong position.
- The optional PEXLST parameter is incorrect or is in the wrong position. Note that the incorrect PEXLST parameter may have come from a command list such as PURGEDB, which invokes NLDM PURGE.

System Action: The NLDM PURGE command is not processed.

Operator Response: Refer to the NetView online help for the NLDM PURGE command or the command list that invokes NLDM PURGE.

AAU927I modid locid INVALID PURGE EXCEPTION LIST xx

Explanation: This message is sent to the authorized message receiver. A missing or invalid PURGE exception list was specified on the NLDM PURGE command or on a command list such as PURGEDB which invokes NLDM PURGE.

Message Variables:
modid The session monitor module where the error condition occurred.
locid The location within the module where the error condition occurred. Possible values include:
 01 Common global variable CNMSTYLE.NLDM.PEXLSTxx.0 (number of entries for this list) could not be accessed.
 02 Common global variable CNMSTYLE.NLDM.PEXLSTxx.0 (number of entries for this list) did not have a numeric value.
 03 Common global variable CNMSTYLE.NLDM.PEXLSTxx.n (one of the entries for this list) could not be accessed.
 04 An entry is not valid (for example, the first token exceeds 9 characters or is not found at position 1).
 05 Duplicate or excessive session type entries (for example, SSCP-SSCP) are defined.
 06 A session type entry (for example SSCP-SSCP) contains an extraneous value.
 07 An entry is not valid (for example, there are not two resource specifications, or one entry has an invalid length).
xx The last two characters of the PEXLSTxx parameter that was specified on the NLDM PURGE command. The xx corresponds to the exception list statement defined in CNMSTYLE or its included members (NLDM.PEXLSTxx).

System Action: If you specified the SESSION keyword on the NLDM PURGE command, the command was not processed. If you specified the ALL keyword, routes may have been purged from the session monitor database, but sessions were not.
Note: NLDM PURGE commands issued indirectly by DBAUTO and DBMAINT specify the ALL keyword. For PURGEDB, the RT/SESS keyword corresponds to ALL, while the SESSION keyword means the same thing in both PURGEDB and NLDM PURGE.

Operator Response: Ensure that you specified the correct PURGE exception list.

System Programmer Response: Verify that the specified PURGE exception list is correct. Check the corresponding definitions in CNMSTYLE or its included members (NLDM.PEXLSTxx).

BNH772I NUMBER OF CONNECTIONS: numcon, MISSED BUFFERS: missbuf

Explanation: This is a multiline message written in response to the TCPCONN QUERY command. It consists of one header line followed by a variable number of data lines, each describing a TCP/IP connection. The number of connections returned is determined by the MAXRECS operand on the TCPCONN QUERY command. This message is designed for parsing by REXX and not for reading by a person. A REXX sample, CNMSTCPC, is provided to format the BNH772I information for readability.

Note: TCPCONN QUERY will display all of the available data, but some error conditions may cause some data to be missing. See any preceding error messages for possible explanations for missing data.

Each data line includes the following information:

Columns

Description

1–8 The TCP/IP name.

10–72 The local IP address. If LADDR in the TCPCONN QUERY was specified using one or more colons (:) this address is output in the form hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh, where each hhh is a 0-to-4-digit hex number. If LADDR was specified using dotted-decimals, this address is output in the form ddd.ddd.ddd.ddd if it is an IP V4 address, or the form ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd if it is an IP V6 address. In either case, ddd is a decimal number from 0 to 255. Otherwise, this address is output in the form ddd.ddd.ddd.ddd if it is an IP V4 address, or the form hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh if it is an IP V6 address. Leading zeros are stripped from output dotted-decimal addresses but not colon-delimited addresses. Unused columns on the right are blank.

74–78 The local port number (right-justified with leading zeros omitted).

80–142 The remote IP address, with formatting handled in the same way as the local IP address (see above).

144–148 The remote port number (right-justified with leading zeros omitted).

150–157 The date and time the connection started, formatted as an eight-character binary store-clock value. This value can be converted into a printable format using the PIPE EDIT command.

The following columns are included only for inactive connections.

159–166 The date and time the connection ended, formatted as an eight-character binary store-clock value. This value can be converted into a printable format using the PIPE EDIT command.

168–175 The socket resource name (the name of the address space that established the connection).

177–184 The eight-digit hexadecimal representation of the local owner task’s TCB address.

186–201 The 16-digit hexadecimal representation of the TCP/IP socket resource ID.

203–214 The number of bytes received over the connection (right-justified, with leading zeros omitted).

216–227 The number of bytes sent over the connection (right-justified, with leading zeros omitted).
The maximum send window size (right-justified, with leading zeros omitted).

The retransmit count (right-justified, with leading zeros omitted).

The protocol indicator (LINEMODE, TN3270, TN3270E, or N/A).

The LU name, if reported by the Communications Server.

The target application name, if reported by the Communications Server.

The logmode name, if reported by the Communications Server.

**Message Variables:**

- **numcon** The total number of connections matching the input criteria. If this number exceeds MAXRECS, the COUNT=YES operand must be specified on the TCPCONN QUERY command in order for this value to be displayed. Otherwise, numcon is N/A.

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

**System Action:** The TCPCONN QUERY is complete.
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, VTAM, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.