Contents

Preface ..................................................... xi
Who Should Read This Document .............................. xi
What This Document Contains ............................... xi
Publications .............................................. xii
   Prerequisite and Related Documents ...................... xii
   Accessing Publications Online ............................ xii
   Ordering Publications .................................... xii
   Providing Feedback about Publications ...................... xiii
Contacting Customer Support ................................ xiii
Accessibility Information .................................... xiii
   Keyboard Access ........................................ xiii
Conventions Used in This Document ......................... xiii
   Platform-specific Information ............................. xiv
   Terminology ............................................. xiv
Reading Syntax Diagrams ..................................... xv
   Required Syntax ......................................... xv
   Optional Keywords and Variables .......................... xv
   Default Values ........................................... xv
   Long Syntax Diagrams ..................................... xv
   Syntax Fragments ......................................... xv
   Commas and Parentheses ................................... xvii
   Highlighting, Brackets, and Braces ........................ xviii
   Abbreviations ............................................ xix

Chapter 1. Location of Statements and Samples .......................... 1
How to Use the NetView Definition Statement Reference ............... 1
Statement Formats ........................................... 6
Syntax Conventions for Definition Statements ...................... 6

Chapter 2. NetView Definition Statement Reference ........................ 9
A (Alert) ................................................... 9
ACCESS ................................................ 11
ACTION ............................................... 12
   AlertRcvName .......................................... 13
   ALIASMEM ............................................ 13
   AMELINIT ............................................ 14
   ASSIGN ............................................. 14
   AUTH ............................................... 15
   AUTOCMD .......................................... 20
   AUTOTASK .......................................... 21
   auxInitCmd .......................................... 22
   banner .............................................. 22
C (Command List) ........................................... 23
   CCDEF ............................................. 23
   CMDCLASS .......................................... 24
   CMDLINE .......................................... 24
   CMDMDL ............................................ 25
   CMDFCVR .......................................... 29
   CMDSYN ............................................ 30
   CNMI .............................................. 31
   COLUMNHEAD ....................................... 31
   COMMON ........................................... 32
   COMNTESC .......................................... 33
   COS .............................................. 34
   DB2SEC ............................................ 34
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLDM.LUCOUNT</td>
</tr>
<tr>
<td>NLDM.MACRF</td>
</tr>
<tr>
<td>NLDM.MAXEND</td>
</tr>
<tr>
<td>NLDM.NETID</td>
</tr>
<tr>
<td>NLDM.PDDNM</td>
</tr>
<tr>
<td>NLDM.PERFMEM</td>
</tr>
<tr>
<td>NLDM.PIUTNUM</td>
</tr>
<tr>
<td>NLDM.PIUTSIZE</td>
</tr>
<tr>
<td>NLDM.PURGE</td>
</tr>
<tr>
<td>NLDM.RETRY</td>
</tr>
<tr>
<td>NLDM.RTDASD</td>
</tr>
<tr>
<td>NLDM.RTIM</td>
</tr>
<tr>
<td>NLDM.RTMDISP</td>
</tr>
<tr>
<td>NLDM.SAW</td>
</tr>
<tr>
<td>NLDM.SAWNUM</td>
</tr>
<tr>
<td>NLDM.SAWSIZE</td>
</tr>
<tr>
<td>NLDM.SDDNNM</td>
</tr>
<tr>
<td>NLDM.SESSION</td>
</tr>
<tr>
<td>NLDM.TRACEGW</td>
</tr>
<tr>
<td>NLDM.TRACELU</td>
</tr>
<tr>
<td>NLDM.TRACESEC</td>
</tr>
<tr>
<td>NMCSTATUS (Control File Entry)</td>
</tr>
<tr>
<td>NOPREFIX</td>
</tr>
<tr>
<td>NORMAL</td>
</tr>
<tr>
<td>NORMQMAX</td>
</tr>
<tr>
<td>NPDA.ALCACHE</td>
</tr>
<tr>
<td>NPDA.ALERTFWD</td>
</tr>
<tr>
<td>NPDA.ALERTLOG</td>
</tr>
<tr>
<td>NPDA.ALRTINFP</td>
</tr>
<tr>
<td>NPDA.ALT_ALERT</td>
</tr>
<tr>
<td>NPDA.AUTO_RATE</td>
</tr>
<tr>
<td>NPDA.DSRBO</td>
</tr>
<tr>
<td>NPDA.DSRBU</td>
</tr>
<tr>
<td>NPDA.ERR_RATE</td>
</tr>
<tr>
<td>NPDA.IHTHRESH</td>
</tr>
<tr>
<td>NPDA.IQTHRESH</td>
</tr>
<tr>
<td>NPDA.MACRF</td>
</tr>
<tr>
<td>NPDA.MDSIND</td>
</tr>
<tr>
<td>NPDA.PDDNM</td>
</tr>
<tr>
<td>NPDA.PNA</td>
</tr>
<tr>
<td>NPDA.PRELOAD</td>
</tr>
<tr>
<td>NPDA.R (Ratio)</td>
</tr>
<tr>
<td>NPDA RATE</td>
</tr>
<tr>
<td>NPDA.REPORTS</td>
</tr>
<tr>
<td>NPDA.SDDNNM</td>
</tr>
<tr>
<td>NPDA.TECROUTE</td>
</tr>
<tr>
<td>NPDA.W (Wrap)</td>
</tr>
<tr>
<td>NQNSUP</td>
</tr>
<tr>
<td>NRM.CMODE</td>
</tr>
<tr>
<td>NRM.CONNINT</td>
</tr>
<tr>
<td>NRM.CONNRETRY</td>
</tr>
<tr>
<td>NRM.HBRETRY</td>
</tr>
<tr>
<td>NRM.HOSTDEST</td>
</tr>
<tr>
<td>NRM.PORT</td>
</tr>
<tr>
<td>NRM.RESET</td>
</tr>
<tr>
<td>NRM.SAMPLERATE</td>
</tr>
<tr>
<td>NRM.STATUS</td>
</tr>
<tr>
<td>NRM.TYPES</td>
</tr>
<tr>
<td>NRM.XCLDOM</td>
</tr>
<tr>
<td>NRM.XCLSRC</td>
</tr>
</tbody>
</table>
Chapter 3. Automated Operations Network Definitions. ........................................... 209
Defining AON Control File Entries ........................................................................ 209
ACTMON .................................................................................................................. 209
ADJNETV .................................................................................................................. 214
AUTOOPS .................................................................................................................. 214
CDLOG ....................................................................................................................... 219
community_name (FKXSCM) ................................................................................. 220
CPCPSESS (SNA) ...................................................................................................... 221
DDFGENERIC .......................................................................................................... 222
DDFGROUP .................................................................................................................. 223
ENVIRON AIP ............................................................................................................. 224
ENVIRON CONSOLE ................................................................................................. 225
ENVIRON DDF ........................................................................................................... 226
ENVIRON EXIT .......................................................................................................... 228
ENVIRON RACF ......................................................................................................... 229
ENVIRON SETUP ....................................................................................................... 231
ENVIRON SNBU (SNA) ............................................................................................... 234
ENVIRON TIMEOUT ................................................................................................. 235
EZLTLOG ..................................................................................................................... 236
FORWARD FOCALPT ................................................................................................. 238
FULLSESS ................................................................................................................... 239
GATEWAY ................................................................................................................. 239
INSTALLOPT ............................................................................................................ 242
IPCONN ...................................................................................................................... 243
IPHOST ...................................................................................................................... 245
IPINFC ......................................................................................................................... 246
IPNAMESERV ........................................................................................................... 248
IPPORT ....................................................................................................................... 250
IPROUTER .................................................................................................................. 253
IPTN3270 .................................................................................................................... 255
LSTHRESH ................................................................................................................. 257
MONIT ......................................................................................................................... 258
MONITOR .................................................................................................................... 259
NCPRRECov (SNA) .................................................................................................... 260
NOTIFY ....................................................................................................................... 263
NTFYOP ...................................................................................................................... 264
NV6000 (TCP/IP) ...................................................................................................... 267
RECOVERY .................................................................................................................. 269
RESIDENT ................................................................................................................... 274
SESSION .................................................................................................................... 274
SNBU (SNA) ................................................................................................................ 275
SNBUPOOL (SNA) ..................................................................................................... 278
SUBSYSTEM (SNA) .................................................................................................... 279
TCP390 ....................................................................................................................... 279
TCPIP ......................................................................................................................... 282
TGWITCH (SNA) ....................................................................................................... 283
THRESHOLDS ............................................................................................................. 284
TIMER .......................................................................................................................... 287
TSOSERV .................................................................................................................... 291
X25MONIT (SNA) ...................................................................................................... 291
### Chapter 4. Inform Policy Member ........................................... 295
- SETUP ............................................................................. 295
- GROUP .......................................................................... 298
- INFORM ......................................................................... 298
- CONTACT ........................................................................ 299

### Chapter 5. Event/Automation Service Definition Statements ................ 305
- AdapterCdsFile. .............................................................. 305
- AdapterFmtFile. .............................................................. 306
- ALRTCFG .................................................................... 307
- ALRTTCFG ................................................................... 308
- BufEvtMaxSize .............................................................. 309
- BufEvtPath .................................................................... 309
- BufEvtRdBlklen .............................................................. 310
- BufEvtShrinkSize .............................................................. 310
- BufferEvents. ................................................................ 311
- BufferEventsLimit ............................................................. 311
- BufferFlushRate .............................................................. 312
- Community .................................................................. 312
- ConnectionMode ............................................................. 312
- Enterpriseoid ................................................................. 313
- ERCVCFG .......................................................... 314
- EventMaxSize ................................................................. 315
- Filter .......................................................................... 315
- FilterCache ................................................................... 316
- FilterMode .................................................................... 317
- Hostname ..................................................................... 317
- MSGCFG ..................................................................... 318
- NetViewAlertReceiver .................................................. 319
- NOSTART .................................................................... 319
- OUTPUT ....................................................................... 320
- PortNumber. .................................................................. 321
- PPI ............................................................................... 321
- RetryInterval ................................................................ 322
- ServerLocation .............................................................. 322
- ServerPort .................................................................... 323
- TALRTCFG ................................................................. 324
- TestMode .................................................................... 325
- TRACE ......................................................................... 325
- UsePortMapper ............................................................... 326

### Chapter 6. Resource Object Data Manager Definition Statements .......... 329
- ASYNC_TASKS ............................................................... 329
- CELL_POOLS ............................................................... 329
- CHARACTER_VALIDATION ............................................... 331
- CHECKPOINT_FUNCTION .............................................. 332
- CONCURRENT_USERS .................................................... 333
- CONNECT_VIOLATION_MESSAGE .................................... 333
- DUMP_FOR_BAD_USER_DATA ......................................... 334
- DUMP_LIMIT ............................................................... 334
- DUMP_SCOPE ............................................................... 335
- EXTEND_HEAP_SIZE ....................................................... 336
- IO_QUEUE_THRESHOLD ................................................... 336
- LOG_LEVEL ................................................................. 337
- MAX_CHUNK .................................................................. 337
- MAX_SEGMENT_NUM ......................................................... 338
- MAX_WINDOW_NUM ......................................................... 339
- MLOG_LEVEL ............................................................... 339
- MTRACE_TYPE .............................................................. 340
- PLI_ISA ....................................................................... 340
PRIMARY_HEAP_SIZE .......................................................... 341
QUIES_WAIT_TASK_TIME ..................................................... 342
QUIES_WAIT_USER_TIME ...................................................... 342
SEC_CLASS ............................................................................. 343
SEC_RNAME .......................................................................... 344
SEGMENT_POCKETS .............................................................. 345
SSB_CHAIN .............................................................................. 346
TRANSPARENT_CHECKPOINT .................................................. 346
WAIT_ALLOC_TIME .............................................................. 347
WINDOW_CHKPT_TIME ............................................................ 348
WINDOW_POCKETS ............................................................... 348

Chapter 7. Graphic Monitor Facility Host Subsystem Statements ................................................. 351
API .......................................................................................... 351
CHECKPOINT ........................................................................... 352
DOMAIN .................................................................................. 353
GMTOFFSET ............................................................................ 353
JAPANESE ................................................................................ 354
LCON-AGG-BUNDLE-INTERVAL .............................................. 354
LCON-AGGRST-REQUIRED .................................................... 355
LCON-AIP-RESET-INTERVAL .................................................. 355
LCON-ALERT-CMD-TIMEOUT .................................................. 356
LCON-ASSOCIATE-NULL-NODE-WITH-LINK .............................. 356
LCON-EVCHANGE-BUFFER-INTERVAL ..................................... 357
LCON-HEX-SUBVECTOR-DISPLAY .......................................... 357
LCON-MAX-LOCATE-RESOURCE-VIEWS .................................... 358
LCON-MAX-QUEUE-DBSERVER ................................................. 358
LCON-MAX-QUEUE-EVENTMGR ............................................... 359
LCON-MAX-QUEUE-IPC ........................................................... 359
LCON-MAX-QUEUE-IRMG R ..................................................... 360
LCON-MAX-QUEUE-MAINTASK .............................................. 361
LCON-MAX-QUEUE-NETCMD .................................................. 361
LCON-MAX-QUEUE-NETCON ................................................... 362
LCON-MAX-QUEUE-OPERIF ..................................................... 362
LCON-MAX-QUEUE-RCMGR ..................................................... 363
LCON-MAX-QUEUE-RTMGR ..................................................... 363
LCON-MAX-QUEUE-VIEWMGR ................................................. 364
LCON-MAX-QUEUE-VSTATMGR ............................................... 364
LCON-NCC-RETRY-LIMIT ....................................................... 365
LCON-NCC-RSC-LIMIT .......................................................... 366
LCON-NMG-POLL-INTERVAL ................................................... 366
LCON-OPERATOR-CMD-AUDIT ............................................... 367
LCON-REPORT-UNKNOWN-STATUS ......................................... 368
LCON-SNATM-TIMEOUT .......................................................... 368
LCON-STATUS-DELAY-MAX ..................................................... 369
LCON-STATUS-DELAY-TIME ................................................... 370
LEVEL ...................................................................................... 370
PRINTPDU38 ........................................................................... 371
RODMID .................................................................................. 372
RODMNAME ........................................................................... 373
STORAGE ................................................................................ 373
TASK ...................................................................................... 374
TRACE ..................................................................................... 374
TRACEBYTES ........................................................................ 375
TRACEPAGES ......................................................................... 376
TYPE ....................................................................................... 376

Index ..................................................................................... 379
Preface

Tivoli® NetView® for z/OS™ enables you to manage complex, multivendor networks and systems from a single point. This book describes the NetView program definition statements required for system administration. One process of system administration is redefining system defaults and storage requirements to improve network performance.

Note: Any reference to MVS™ in this book applies to OS/390® or z/OS, unless otherwise specified.

Who Should Read This Document

This book is a reference source for system programmers and network operators who need a complete understanding of the NetView program definition statements.

What This Document Contains

This document is organized into the following chapters:

- “Chapter 1. Location of Statements and Samples” on page 1 provides a table that explains the location of statements within samples and location of samples within the NetView program. It also describes the format of the definition statements provided in this book.
- “Chapter 2. NetView Definition Statement Reference” on page 9 contains an alphabetical listing of all NetView definition statements. Each entry contains an explanation of the purpose of the statement and identifies the member where it is used. The format of each entry provides all operands, default values, and information that must be supplied in the statement.
- “Chapter 3. Automated Operations Network Definitions” on page 209 contains an alphabetical listing of all AON definition statements. Each entry contains an explanation of the purpose of the statement and identifies the member where it is used. The format of each entry provides all operands, default values, and information that must be supplied in the statement.
- “Chapter 4. Inform Policy Member” on page 293 defines which personnel should be contacted, when they should be contacted and how they should be contacted.
- “Chapter 5. Event/Automation Service Definition Statements” on page 305 provides an alphabetical listing of the statements needed to define the Event/Automation task.
- “Chapter 6. Resource Object Data Manager Definition Statements” on page 329 provides an alphabetical listing of the definition statements needed to define the Resource Object Data Manager (RODM). It also provides an alphabetical listing of the definition statements needed to define the RODM automation task.
- “Chapter 7. Graphic Monitor Facility Host Subsystem Statements” on page 351 provides an alphabetical listing of the initialization statements needed to define the NetView Graphic Monitor Facility host subsystem (GMFHS).
This section lists prerequisite and related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to make comments on Tivoli publications.

**Prerequisite and Related Documents**

To read about the new functions offered in this release, refer to the *Tivoli NetView for z/OS Installation: Migration Guide*.

You can find additional product information on these Internet sites:

<table>
<thead>
<tr>
<th>Table 1. Resource Web sites</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tivoli NetView for z/OS</td>
<td><a href="http://www.tivoli.com/nv390">http://www.tivoli.com/nv390</a></td>
</tr>
</tbody>
</table>

The Tivoli NetView for z/OS Web site offers demonstrations of the NetView product, related products, and several free NetView applications you can download. These applications can help you with tasks such as:

- Getting statistics for your automation table and merging the statistics with a listing of the automation table
- Displaying the status of a JES job or cancelling a specified JES job
- Sending alerts to the NetView program using the program-to-program interface (PPI)
- Sending and receiving MVS commands using the PPI
- Sending TSO commands and receiving responses

**Accessing Publications Online**

You can access many Tivoli publications online using the Tivoli Information Center, which is available on the Tivoli Customer Support Web site:


These publications are available in PDF format. Translated documents are also available for some products.

**Ordering Publications**

You can order many Tivoli publications online at the following Web site:


You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968
- In other countries, for a list of telephone numbers, see the following Web site: [http://www.tivoli.com/inside/store/lit_order.htm](http://www.tivoli.com/inside/store/lit_order.htm)
Providing Feedback about Publications

We are very interested in hearing about your experience with Tivoli products and documentation, and we welcome your suggestions for improvements. If you have comments or suggestions about our products and documentation, contact us in one of the following ways:

- Send an e-mail to pubs@tivoli.com.
- Complete our customer feedback survey at the following Web site: 

Contacting Customer Support

If you have a problem with any Tivoli product, you can contact Tivoli Customer Support. See the Tivoli Customer Support Handbook at the following Web site:


The handbook provides information about how to contact Tivoli Customer 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 you are in
- What information you should gather before contacting support

Note: Additional support for Tivoli NetView for z/OS is available at the NetView for z/OS home page:

[http://www.tivoli.com/nv390](http://www.tivoli.com/nv390) Under Related Documents, select Other Online Sources. The page displayed contains a list of newsgroups, forums, and bulletin boards.

Accessibility Information

Refer to Tivoli NetView for z/OS User’s Guide for information about accessibility.

Keyboard Access

Standard shortcut and accelerator keys are used by the product and are documented by the operating system. Refer to the documentation provided by your operating system for more information.

Refer to Tivoli NetView for z/OS User’s Guide for more information about keyboard access.

Conventions Used in This Document

The document uses several typeface conventions for special terms and actions. These conventions have the following meaning:

- **Bold** Commands, keywords, flags, and other information that you must use literally appear like this, in **bold**.
- **Italics** Variables and new terms appear like this, in *italics*. Words and phrases that are emphasized also appear like this, in *italics*.
- **Monospace** Code examples, output, and system messages appear like this, in a monospace font.
ALL CAPS  Tivoli NetView for z/OS commands are in ALL CAPITAL letters.

Platform-specific Information

For more information about the hardware and software requirements for NetView components, refer to the Tivoli NetView for z/OS Licensed Program Specification.

Terminology

For a list of Tivoli NetView for z/OS terms and definitions, refer to http://www.networking.ibm.com/nsg/nsgmain.htm.

For brevity and readability, the following terms are used in this document:

NetView
- Tivoli NetView for z/OS Version 5 Release 1
- Tivoli NetView for OS/390 Version 1 Release 4
- Tivoli NetView for OS/390 Version 1 Release 3
- TME 10™ NetView for OS/390 Version 1 Release 2
- TME 10 NetView for OS/390 Version 1 Release 1
- IBM NetView for MVS Version 3
- IBM NetView for MVS Version 2 Release 4
- IBM NetView Version 2 Release 3

MVS  OS/390 or z/OS operating systems.

RACF®
RACF is a component of the SecureWay® Security Server for z/OS and OS/390, providing the functions of authentication and access control for OS/390 and z/OS resources and data, including the ability to control access to DB2® objects using RACF profiles. Refer to:

Tivoli Enterprise™ software
Tivoli software that manages large business networks.

Tivoli environment
The Tivoli applications, based upon the Tivoli Management Framework, that are installed at a specific customer location and that address network computing management issues across many platforms. In a Tivoli environment, a system administrator can distribute software, manage user configurations, change access privileges, automate operations, monitor resources, and schedule jobs. You may have used TME 10 environment in the past.

TME 10
In most product names, TME 10 has been changed to Tivoli.

V and R
Specifies the version and release.

VTAM® and TCP/IP
VTAM and TCP/IP are included in the IBM Communications Server element of the OS/390 and z/OS operating systems. Refer to http://www.ibm.com/software/network/commserver/about/
Unless otherwise indicated, references to programs indicate the latest version and release of the programs. If only a version is indicated, the reference is to all releases within that version.

When a reference is made about using a personal computer or workstation, any programmable workstation can be used.

**Reading Syntax Diagrams**

Syntax diagrams start with double arrowheads on the left (\(\Rightarrow\)) and move along the main line until they end with two arrowheads facing each other (\(\Rightarrow\)).

As shown in the following table, syntax diagrams use *position* to indicate the required, optional, and default values for keywords, variables, and operands.

<table>
<thead>
<tr>
<th>Element Position</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>On the command line</td>
<td>Required</td>
</tr>
<tr>
<td>Above the command line</td>
<td>Default</td>
</tr>
<tr>
<td>Below the command line</td>
<td>Optional</td>
</tr>
</tbody>
</table>

**Required Syntax**

The command name, required keywords, variables, and operands are always on the main syntax line. Figure 1 specifies that the `resname` variable must be used for the CCPLOADF command.

```
Figure 1. Required Syntax Elements

CCPLOADF
\(\Rightarrow\)CCPLOADF resname\(\Rightarrow\)
```

Keywords and operands are written in uppercase letters. Lowercase letters indicate variables such as values or names that you supply. In Figure 2, MEMBER is an operand and `membername` is a variable that defines the name of the data set member for that operand.

```
Figure 2. Syntax for Variables

TRANSMMSG
\(\Rightarrow\)TRANSMMSG MEMBER=membername\(\Rightarrow\)
```

**Optional Keywords and Variables**

Optional keywords, variables, and operands are below the main syntax line. Figure 3 on page xvi specifies that the ID operand can be used for the DISPREG command, but is not required.
Default Values

Default values are above the main syntax line. If the default is a keyword, it appears only above the main line. You can specify this keyword or allow it to default.

If an operand has a default value, the operand appears both above and below the main line. A value below the main line indicates that if you choose to specify the operand, you must also specify either the default value or another value shown. If you do not specify an operand, the default value above the main line is used.

Figure 4 shows the default keyword STEP above the main line and the rest of the optional keywords below the main line. It also shows the default values for operands MODNAME=* and OPTION=* above and below the main line.

RID

Figure 4. Sample of Defaults Syntax

Long Syntax Diagrams

When more than one line is needed for a syntax diagram, the continued lines end with a single arrowhead (►). The following lines begin with a single arrowhead (►), as shown in Figure 4.

Syntax Fragments

Commands that contain lengthy groups or a section that is used more than once in a command are shown as separate fragments following the main diagram. The fragment name is shown in mixed case. See Figure 5 on page xvii for a syntax with the fragments ReMote and FromTo.
Commas and Parentheses

Required commas and parentheses are included in the syntax diagram. When an operand has more than one value, the values are typically enclosed in parentheses and separated by commas. In Figure 6 on page xviii, the OP operand, for example, contains commas to indicate that you can specify multiple values for the testop variable.
If a command requires positional commas to separate keywords and variables, the commas are shown before the keyword or variable, as in Figure 4 on page xvi.

For example, to specify the BOSESS command with the sessid variable, enter:

```
NCCF BOSESS applid,,sessid
```

You do not need to specify the trailing positional commas. Positional and non-positional trailing commas either are ignored or cause the command to be rejected. Restrictions for each command state whether trailing commas cause the command to be rejected.

**Highlighting, Brackets, and Braces**

Syntax diagrams do not rely on highlighting, underscoring, brackets, or braces; variables are shown italicized in hardcopy or in a differentiating color for NetView help and BookManager® online books.

In parameter descriptions, the appearance of syntax elements in a diagram immediately tells you the type of element. See Table 3 for the appearance of syntax elements.

**Table 3. Syntax Elements Examples**

<table>
<thead>
<tr>
<th>This element...</th>
<th>Looks like this...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>CCPLOADF</td>
</tr>
<tr>
<td>Variable</td>
<td>resname</td>
</tr>
<tr>
<td>Operand</td>
<td>MEMBER=membername</td>
</tr>
<tr>
<td>Default</td>
<td>today or INCL</td>
</tr>
</tbody>
</table>

Figure 6. Sample Syntax Diagram with Commas

If a command requires positional commas to separate keywords and variables, the commas are shown before the keyword or variable, as in Figure 4 on page xvi.

For example, to specify the BOSESS command with the sessid variable, enter:

```
NCCF BOSESS applid,,sessid
```

You do not need to specify the trailing positional commas. Positional and non-positional trailing commas either are ignored or cause the command to be rejected. Restrictions for each command state whether trailing commas cause the command to be rejected.
Abbreviations

Command and keyword abbreviations are described in synonym tables after each command description.
Chapter 1. Location of Statements and Samples

This chapter provides a table that explains the location of statements within samples and the location of samples within the NetView program.

How to Use the NetView Definition Statement Reference

Use the following methods to locate specific definition statement information:

- When you know a statement or keyword name, look it up in alphabetical order. The definition statements are listed in alphabetical order in their respective chapters. Use the dictionary-style headings for quick reference.
- When you know an operand name but not the statement with which it is used, locate the operand name in the index.
- When you know only the sample name and want information on the statements in the sample, use Table 4 to find the sample name and its associated statements.

Table 4. Location of Statements within Samples and Location of Samples within the NetView Program

<table>
<thead>
<tr>
<th>Sample</th>
<th>Statements</th>
<th>Page</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAUCNMTD</td>
<td>DSTINIT</td>
<td>41</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>AAUPRMLP</td>
<td>All statements are defined in CNMSTYLE and referenced by AAUPRMLP.</td>
<td>41</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>A01APPLS</td>
<td>STATOPT</td>
<td>185</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>A01SNA</td>
<td>STATOPT</td>
<td>185</td>
<td>VTAMLST</td>
</tr>
<tr>
<td>BNJMBDST</td>
<td>All statements are defined in CNMSTYLE and referenced by BNJMBDST.</td>
<td></td>
<td>DSIPARM</td>
</tr>
<tr>
<td>BNJ36DST</td>
<td>DSTINIT</td>
<td>41</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>CNMSCNFT</td>
<td>%INCLUDE</td>
<td>55</td>
<td>DSIPARM</td>
</tr>
<tr>
<td></td>
<td>ACTION</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AuxInitCmd</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CMDLINE</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COLUMNHEAD</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HELD</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOLDPCNT</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HOLDWARN</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IMDAREA</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INDENT</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LASTLINE</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOCKIND</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MLINDENT</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NOPREFIX</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NORMAL</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NORMQMAX</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PREFIX</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLE</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLEDATE</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLEDOMID</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLOPID</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLESTAT</td>
<td>117</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TITLETEM</td>
<td>117</td>
<td></td>
</tr>
</tbody>
</table>
### Table 4. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

<table>
<thead>
<tr>
<th>Sample</th>
<th>Statements</th>
<th>Page</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMSTYLE</td>
<td>AlertRcvName</td>
<td>14</td>
<td>DSIPARM</td>
</tr>
<tr>
<td></td>
<td>ASSIGN</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUTOCMD</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AUTOTASK</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>banner</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCDEF</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CNMI</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>COMMON</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DB2SEC</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DBFULL</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DEFAULTS</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DOMAIN</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DSTINIT</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td></td>
<td>function.autotask</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GHBTCPANAME</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HARDCOPY</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HLL</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IdleParms</td>
<td>55</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INIT</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td></td>
<td>inStore</td>
<td>52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IPLOG</td>
<td>57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JesJobLog</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOGONPW</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LUC</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MCON</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MemStore</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MVSPARM</td>
<td>81</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NetID</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NLDLM</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NMCstatus</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NPDA</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NRM</td>
<td>139</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OpDsPrefix</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td></td>
<td>POLICY</td>
<td>157</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RCMRODMUser</td>
<td>166</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REXEC</td>
<td>165</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RMTINIT</td>
<td>166</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RODMname</td>
<td>166</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RSH</td>
<td>171</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTT</td>
<td>172</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SECOPTS</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSIName</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td></td>
<td>styleMsg</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SUPPCHAR</td>
<td>188</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TAMEL</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TASK</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCPname</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCPserver</td>
<td>190</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TOWER</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRACE</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>transMember</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>transTbl</td>
<td>203</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VBV</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td></td>
<td>VTAMCP</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WEB</td>
<td>208</td>
<td></td>
</tr>
<tr>
<td>DSIALATD</td>
<td>ALIASMEM</td>
<td>18</td>
<td>DSIPARM</td>
</tr>
<tr>
<td></td>
<td>DSTINIT</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Sample</td>
<td>Statements</td>
<td>Page Location</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>DSIALTAB</td>
<td>COS, LU, MODE, ORIGNET</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIAMLD</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIAMLD.</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSICMD</td>
<td>%INCLUDE, CMDMDL, CMDSYN, COMNTESC, END, PARMSYN</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSICNM</td>
<td>A (Alert), C (Command List), F (Filter), M (Maximum), O MONIT, O RESET, O SECSTAT, O SENDMSG, SENDMSG</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSICPIN</td>
<td>DSTINIT, END</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSICRTTD</td>
<td>DEFFOCPT, DSTINIT, END</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIDB2DF</td>
<td>SUBSYSTEM</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIELMEM</td>
<td>DSTINIT</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIELXIT</td>
<td>END</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIIHINIT</td>
<td>DSTINIT, PARTNER, RETRY</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIIILGCF</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIIILGCF.</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIKINIT</td>
<td>DSTINIT</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSILOGBK</td>
<td>All statements are defined in CNMSTYLE and referenced by DSILOGBK.</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSILUCTD</td>
<td>All statements are defined in CNMSTYLE and referenced by DSILUCTD.</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIOFF</td>
<td>%INCLUDE, END, OPERATOR, PROFILEN</td>
<td>DSIPARM</td>
<td></td>
</tr>
<tr>
<td>DSIPRFxx</td>
<td>AUTH, DOMAINS, END, ISPAN, PROFILEN, SPAN</td>
<td>DSIPARM</td>
<td></td>
</tr>
</tbody>
</table>
Table 4. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

<table>
<thead>
<tr>
<th>Sample</th>
<th>Statements</th>
<th>Page</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIPROFx</td>
<td>AUTH, DOMAINS, END, ISPAN, PROFILE, SPAN</td>
<td>15</td>
<td>DSIPRF</td>
</tr>
<tr>
<td>DSIQTSKI</td>
<td>CMDRCVR, REP, TASK</td>
<td>28</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIREXCF</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIREXCF.</td>
<td>157</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRHOST</td>
<td>ACCESS</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIROVSI</td>
<td>DSTINIT, PUCOUNT</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRSHCF</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIRSHCF.</td>
<td>157</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIRTITD</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIRTITD.</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSISECUR</td>
<td>RMTSEC</td>
<td>157</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIVVRTO</td>
<td>DSTINIT</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCPCF</td>
<td>All statements are defined in CNMSTYLE and referenced by DSITCPCF.</td>
<td>157</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCPRF</td>
<td>OPERID</td>
<td>151</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSITCBK</td>
<td>DSTINIT, LOGINIT</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIVIPARM</td>
<td>DSTINIT, VPDINIT</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIBMEM</td>
<td>All statements are defined in CNMSTYLE and referenced by DSIBMEM.</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DSIBINIT</td>
<td>DEFENTPT, DEFFOCPT, DSTINIT, RETRY</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
<tr>
<td>DUIFPMEM</td>
<td>All statements are defined in CNMSTYLE and referenced by DUIFPMEM.</td>
<td>11</td>
<td>DSIPARM</td>
</tr>
</tbody>
</table>
Table 4. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

<table>
<thead>
<tr>
<th>Sample</th>
<th>Statements</th>
<th>Page</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUIGINIT</td>
<td>API</td>
<td>351</td>
<td>DSIPARM</td>
</tr>
<tr>
<td></td>
<td>CHECKPOINT</td>
<td>352</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DOMAIN</td>
<td>353</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GMTOFFSET</td>
<td>354</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-AGG-BUNDLE-INTERVAL</td>
<td>355</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-AGGRST-REQUIRED</td>
<td>356</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-ALERT-CMD TIMEOUT</td>
<td>357</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-AIP-RESET-INTERVAL</td>
<td>358</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-ASSOCIATE-NULL-NODE-WITH-LINK</td>
<td>359</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-EVCHANGE-BUFFER-INTERVAL</td>
<td>360</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-HEX-SUBVECTOR-DISPLAY</td>
<td>361</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-LOCATE-RESOURCE-VIEWS</td>
<td>362</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-DBSERVER</td>
<td>363</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-EVENTMGR</td>
<td>364</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-IPC</td>
<td>365</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-IRMGR</td>
<td>366</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-MAINTASK</td>
<td>367</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-NETCMD</td>
<td>368</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-NETCON</td>
<td>369</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-RCMGR</td>
<td>370</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-RTMGR</td>
<td>371</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-VIEWMGR</td>
<td>372</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-MAX-QUEUE-VSTATMGR</td>
<td>373</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-NCC-RETRY-LIMIT</td>
<td>374</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-NCC-RSC-LIMIT</td>
<td>375</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-NMG-POLL-INTERVAL</td>
<td>376</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-OPERATOR-CMD-AUDIT</td>
<td>377</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-REPORT-UNKNOWN-STATUS</td>
<td>378</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-SNATM-TIMEOUT</td>
<td>379</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-STATUS-DELAY-MAX</td>
<td>380</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LCON-STATUS-DELAY-TIME</td>
<td>381</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEVEL</td>
<td>382</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRINTPDU38</td>
<td>383</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RODMID</td>
<td>384</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RODMNAME</td>
<td>385</td>
<td></td>
</tr>
<tr>
<td></td>
<td>STORAGE</td>
<td>386</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TASK</td>
<td>387</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRACE</td>
<td>388</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRACEBYTES</td>
<td>389</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRACEPAGES</td>
<td>390</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TYPE</td>
<td>391</td>
<td></td>
</tr>
</tbody>
</table>

| DUIGHB    | All statements are defined in CNMSTYLE and referenced by DUIGHB. | 41   | DSIPARM  |

| DUISISFP  | AMELINIT                          | 42   | DSIPARM  |
|           | DSTINIT                            | 43   |          |
### Statement Formats

The format of a definition statement is:

- **Statement name**
- **General introduction**
  - The general introduction explains overall options, assumptions, and the purpose of the statement. Each introduction explains the name of the member and where you code the statement.
- **Definition statement syntax**
  - The definition statement syntax is a model statement that is formatted according to the code conventions.
- **Operand descriptions**
  - This section describes each operand you can specify for the definition statement. The description includes the specific values or variable information that you can specify for the operand.
- **Cross-reference to related statements**
  - This section lists other statements that can affect the definition statement.

### Syntax Conventions for Definition Statements

These syntax conventions apply to most statements:

---

**Table 4. Location of Statements within Samples and Location of Samples within the NetView Program (continued)**

<table>
<thead>
<tr>
<th>Sample</th>
<th>Statements</th>
<th>Page</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>EKGCUST</td>
<td>ASYNC_TASKS</td>
<td>329</td>
<td>CNMSAMP</td>
</tr>
<tr>
<td></td>
<td>CELL_POOLS</td>
<td>329</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHARACTER_VALIDATION</td>
<td>332</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHECKPOINT_FUNCTION</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONCURRENT_USERS</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CONNECT_VIOLATION_MESSAGE</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DUMP_FOR_BAD_USER_DATA</td>
<td>333</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DUMP_LIMIT</td>
<td>334</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DUMP_SCOPE</td>
<td>335</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXTEND_HEAP_SIZE</td>
<td>336</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IO_QUEUE_THRESHOLD</td>
<td>336</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LOG_LEVEL</td>
<td>337</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAX_CHUNK</td>
<td>337</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAX_SEGMENT_NUM</td>
<td>338</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAX_WINDOW_NUM</td>
<td>339</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MLOG_LEVEL</td>
<td>339</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MTRACE_TYPE</td>
<td>340</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PLI_ISA</td>
<td>340</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRIMARY_HEAP_SIZE</td>
<td>341</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QUIES_WAIT_TASK_TIME</td>
<td>342</td>
<td></td>
</tr>
<tr>
<td></td>
<td>QUIES_WAIT_USER_TIME</td>
<td>342</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEC_CLASS</td>
<td>343</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEC_RNAME</td>
<td>344</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SEGMENT_POCKETS</td>
<td>345</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSB_CHAIN</td>
<td>346</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TRANSPARENT_CHECKPOINT</td>
<td>347</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WAIT_ALLOC_TIME</td>
<td>347</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WINDOW_CHKPT_TIME</td>
<td>348</td>
<td></td>
</tr>
<tr>
<td></td>
<td>WINDOW_POCKETS</td>
<td>348</td>
<td></td>
</tr>
</tbody>
</table>
• Code at least one blank between a label name and the name of the definition statement, and between the name of the definition statement and the first operand. One or more blanks, or a single comma with no blanks, must separate the statement operands. You cannot separate the operands with a combination of commas and blanks. If you omit the optional label name, you still need to precede the definition statement with one or more blanks.

• The label field must not exceed 8 characters, and the field must start in column 1.

• Continuation from one line to the next is not allowed. However, you can repeat the definition statement and add the remaining information. For the following example:

  LOGINIT AUTOFLIP=YES
  LOGINIT RESUME=YES

  Is the same as:

  LOGINIT AUTOFLIP=YES,RESUME=YES

• Place comments on a separate line for DSIPARM members. The first column of a comment line must contain an asterisk (*).

• Many definition files conclude with an END statement. This END statement has no operands and cannot begin in column 1.

• All NetView program identifiers, which are called names, must not exceed 8 characters unless specified. The first character must be alphabetical and alphabetical characters must be in uppercase.

• Command names, command list names, and any other NetView program identifiers must not contain commas (,), periods (.), blanks ( ), apostrophes (‘), ampersands (&), asterisks (*), or equal signs (=). Commas, periods, blanks, and equal signs are used as delimiters when the definition statements are parsed. The other characters have special meanings for NetView command lists.

• Command names and command list names must begin in column 1.

• System symbolics can be coded on any NetView definition statement to provide unique information to the NetView system. System symbolics are useful when running NetView on different systems where you want to have different characteristics. This unique information (as defined by the system symbolic values) will remain on your system definitions until you change those definitions and re-IPL MVS.
Chapter 2. NetView Definition Statement Reference

NetView definition statements are used by the NetView program for performing system administration tasks. System administration is the process of redefining system defaults and storage requirements. You can perform administration subtasks during NetView program installation or when you redefine network resources.

In planning for installation and network management tasks, you determine the facilities you need to run the NetView program. You also need to identify hardware requirements and specific resources used by the NetView program. This information can help you determine requirements for user coding and NetView program definitions.

Use the definition statements to accomplish the procedures described in the Tivoli NetView for z/OS Installation: Getting Started. This book introduces the concept of altering the sample NetView program definition statements to suit your requirements. After copying some current definition statements from the sample files, you can begin to modify existing definition statements or create new ones. You can alter definition statements during a first-time NetView program installation, or later, while running your production system.

A (Alert)

The A (Alert) statement defines the color, highlighting, and alarm attributes corresponding to the message indicators in the status monitor. Messages in the network log are also displayed in the specified colors when you browse the network log data. Alarm, color, or highlighting attributes that are defined for a message occur for each operator who is defined to receive the message.

Code the alert statement in DSICNM. DSICNM changes do not take effect until you stop and restart the \texttt{nv_id} VMT task (where \texttt{nv_id} is the NetView domain identifier). If changing the member results in additional extents being used, stop and restart the NetView program before the changes take effect.

\textbf{Note:} Message indicator definitions for a given operator are not enabled until the operator has accessed the status monitor.

Code one A statement for each message indicator class you define. You can code four message indicator class statements (1–4). If you code A statements, use automation table statements to assign messages to the message indicator classes. You can use automation table statements to route these messages to any NetView program operators. Refer to Tivoli NetView for z/OS Automation Guide for more information. You can also use F statements to assign messages to the message indicator classes and, in this case, the messages are always routed to the authorized message receiver.

The syntax for the A statement is:

\begin{verbatim}
A class color notify high alarm console comments
\end{verbatim}
A (Alert)

Where:

A Indicates the statement name, which must be in column 1.

class
Indicates the message indicator class number and must be from 1–4. The message indicator value must be in column 2 of the statement. This number is displayed at the top of any status monitor panel and is associated with a number on an F statement.

color
Indicates the color value for the message indicator class. This value must be in column 4 of the statement. The number at the top of any status monitor panel (when alerted) and the message in the network log are displayed in the chosen color. You can use the following color values:

B Blue
G Green
P Pink
R Red
T Turquoise
W White
Y Yellow
(Blank) Default color for the terminal. This is the default.

notify
Specifies that the alarm and high definitions are triggered for the operators when messages in this class arrive. This value must be in column 5 of the statement. You can use the following notify values:

Y The high and alarm values are interpreted.
N The high and alarm values must be blank. This is the default.

high
Indicates the highlighting value for the specified message indicator class. This value must be in column 6 of the statement. The number at the top of each operator’s status monitor panel is highlighted according to the highlighting value.

You can use the following high values:

B Blink
R Reverse video
U Underscored
(Blank) No highlighting. This is the default.

Note: This value must be blank if notify is N.

alarm
Indicates the alarm value for the message indicator class. This value must be in column 7 of the statement. When the alarm is set, any message from the specified class generates an alarm for each operator when the message arrives. The alarm is sounded only when the operator is in the status monitor. You can use the following alarm values:

Y The alarm is sounded when a message arrives.
N The alarm is not sounded. This is the default.

Note: This value must be blank if notify is N.

console
Indicates whether a copy of the message should be sent to the system console.
This value must be in column 8 of the statement. When you set this value, any message from the specified class is sent to the system console. You can use the following console values:

- **Y**: A copy of the message is sent to the system console.
- **N**: A copy of the message is not sent to the system console. This is the default.

**comments**

Indicates unused space for the statement. Any comments you want included with the statement must be in columns 9–80.

**Related Statements:** F, AUTH

---

**ACCESS**

The ACCESS statement allows you to control remote access to NetView from users of the RSH command. These statements are placed in the DSIRHOST member of DSIPARM. Each statement consists of a hostname and an optional user name on a single line, with one exception. The special character "+" can appear on a line by itself to signal universal access except for users or hosts specifically prohibited by another ACCESS statement.

The syntax for the ACCESS statement is:

```
ACCESS
```

Where:

- **+**: Specifies to allow access to the host or user. It is the default.
- **-**: Specifies to deny access to the host or user.

**host**

Specifies a remote host. It can be specified as a TCP/IP host name or as an IP address in dotted notation (for example: 127.44.44.44).

**user**

Specifies the username on the remote system.

**Usage Notes:**

- Once a host has been specified in an ACCESS statement, subsequent access statements must agree with the security level set for that host. For example, if "-host1" has been specified, "host1" or "+host1" would be invalid on a latter ACCESS statement.
- Specific ACCESS statements for a host/user override the use of the global "+" character for the specified host or user. This means that "+" can be used to set a default access level but particular hosts or users can be excluded.
The ACTION statement enables you to change the color of action messages that appear on the command facility panel. The sample member containing the ACTION statement is CNMSCNFT. You can code the ACTION statement only once.

The syntax for the ACTION statement is:

```
ACTION
```

Where:

- `colorf` defines foreground color for action messages. You must specify the foreground color before the background color.
- `attribute` defines alarm, intensity, and highlight attributes for action messages. You can specify attributes only once for the ACTION statement.
- `ON` makes the color following ON apply to the background of the action messages. This is a required keyword if you specify only a background color.
- `colorb` defines background color for action messages.

Usage Notes:

- To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the ACTION statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm on
- The color operands are:
  - `BLACK` the color black
  - `BLUE` the color blue
  - `RED` the color red
  - `PINK` the color pink
  - `GREEN` the color green
  - `TURQ` the color turquoise
  - `YELLOW` the color yellow
  - `WHITE` the color white
- The highlight operands are:
  - `BLINK` the characters blink.
  - `REV` the characters are changed to reverse video.
  - `UND` the characters are underlined.
  - `NRM` normal attributes are used.
- The intensity operands are:
  - **HIGH** The characters have high intensity.
  - **LOW** The characters have low intensity.

- The alarm operand is:
  - **BEEP** An audible alarm sounds when a message is shown.

**Related Statements:** CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOID, TITLESTAT, TITLETIME

---

**AlertRcvName**

The AlertRcvName statement defines the PPI alert receiver name associated with the MOD=CNMCALRT task. This statement is coded in CNMSTYLE.

The syntax for the AUTOTASK statement is:

```
AlertRcvName
```

Where:

- **name**
  - Specifies a 1–8 character PPI alert receiver name.

**Usage Note:** If you run more than one alert receiver task on this NetView program, specify a 1–8 character name beginning with an asterisk (*). In this case, each PPI name will be set to the task name. For the task name CNMCALRT, the PPI name is always set to NETVALRT.

---

**ALIASMEM**

The ALIASMEM statement creates a table of alias name translations. Code this statement in the member specified by the MEM keyword of the ALIASAPL TASK statement. The sample member supplied with the NetView program is DSIALATD. There is a sample table of alias name translations shipped with the NetView program as DSIPARM member DSIALTAB.

Changes to the member containing the ALIASMEM statements do not take effect until you stop and restart the ALIASAPL task. If changing the member results in additional extents being used, stop and restart the NetView program to implement the changes.

**Note:** Code this statement after the DSTINIT statement that defines DSIAINIT as the DST initialization exit.

The syntax for the ALIASMEM statement is:

```
ALIASMEM
```
Where:

**label**
Indicates the optional label for the ALIASMEM statement. This label identifies the statement in any related error messages.

**membername** [,...]
Names the members that contain the ORIGNET, LU, COS, and MODE definition statements.

**Related Statements:** COS, DSTINIT, LU, MODE, ORIGNET, TASK

---

**AMELINIT**

The AMELINIT statement defines the initialization operands required by the CNMTAMEL data services task (DST). Code the AMELINIT statement in the member specified by the MEM keyword on the CNMTAMEL TASK statement. The sample member provided with the NetView program is DUIISFP.

Changes to DUIISFP do not take effect until you stop and restart the CNMTAMEL task. If changing the member results in using additional extents in the DSIPARM data set, stop and restart the NetView program before the changes take effect. In DUIISFP, use the AMELINIT statement to bring up CNMTAMEL as a status focal point.

The syntax for the AMELINIT statement is:

```
AMELINIT
```

Where:

**DUIFPMEM**
Designates that the initialization operands required by the CNMTAMEL task are in the status focal point initialization member DUIFPMEM.

**memb_name**
Indicates a 1–8 character name of the member containing the initialization operands required by the CNMTAMEL task. You can create this member by copying DUIFPMEM to a new member and customizing that member.

---

**ASSIGN**

You can route messages to groups of operators. Routing groups that appear in message automation must be defined before automation is enabled. The ASSIGN statement identifies the operator groups. This statement is coded in CNMSTYLE.

The syntax for the ASSIGN statement is:
ASSIGN

```
ASSIGN groupname.GROUP=operatorid
```

**Where:**

- `groupname` is the 1–7 character group name.
- `operatorid` is the list of operator IDs separated by commas.

**Usage Note:** You can also use the ASSIGN command to assign operators to groups. For more information, refer to the [Tivoli NetView for z/OS Command Reference](#) or the NetView online help.

---

**AUTH**

The AUTH statement defines an operator’s authority to view and control resources, specifies whether an operator is eligible to be the authorized receiver, defines an operator’s authority to perform NetView management console administrative functions, and specifies whether an operator is allowed to sign on to NetView management console with the Use commands option selected. Code this statement in a member specified by a PROFILEN statement associated with the operator. See “OPERATOR” on page 150 and “PROFILEN” on page 161 for information on how a PROFILEN statement is associated with an operator. A sample member supplied with the NetView program is DSIPROFA.

The syntax for the AUTH statement is:

```
AUTH
```

**NGMFVSPN variables:**
Where:

**label**
Is the optional label for the AUTH statement. This label identifies the statement in related error messages.

**MSGRECVR=NO | YES**
Specifies whether operators using this profile can receive unsolicited messages that are not routed to a particular operator by the use of the NetView ASSIGN command or by NetView automation.

**NO**
Indicates that operators using the profile containing this statement do not receive unsolicited messages. NO is the default.

**YES**
Indicates that an operator using this profile can be the authorized message receiver.

Notes:
1. In the NetView program, the *authorized receiver* is the operator authorized to receive all the unsolicited and authorized messages that are not routed to a specific operator with an ASSIGN command or a ROUTE action in a NetView automation statement. The authorized receiver is determined by the order in which operator terminals are defined to the NetView program and by the order in which authorized operators have logged on.

2. When several operators are eligible to receive a particular message, the NetView program uses the following priority order (from the lowest to the highest) to route the message to the proper operator:
   - The operator designated by an ASSIGN command
   - The operator or operators designated by the ROUTE action in the automation table
   - A cross-domain operator
     If more than one cross-domain operator is logged on, the one logged on first has priority.
   - An autotask operator
     If more than one autotask has been started, the one started first has priority. Use the ASSIGN command if an autotask is going to be the receiver of unsolicited messages.
   - The system console operator

**CTL=SPECIFIC | GENERAL | GLOBAL**
Defines the operator’s authority to control resources. The value of this operand must indicate one of the following:

**SPECIFIC**
An operator can control only the resources and view names that are members of a span to which that operator has authority. These are listed on ISPAN and SPAN statements in the operator’s profile, or if OPSPAN=SADF, in the NETSPAN class of an SAF product, such as RACF. The operator can establish NNT cross-domain sessions only with the NCCFIDs listed on the DOMAINS statement. SPECIFIC is the default value.
GENERAL
An operator can control the resources and view names that are members of a span to which that operator has authority. These are listed on ISPAN and SPAN statements in the operator’s profile, or if OPSPAN=SAF, in the NETSPAN class of an SAF product, such as RACF. The operator can also control resources that do not belong to any span. The operator can establish NNT cross-domain sessions only with the NCCFIDs listed on the DOMAINS statement.

GLOBAL
Span of control is not used. DOMAINS, ISPAN, and SPAN statements and the span names in the NETSPAN class of an SAF product are not used. An operator with global authority can establish NNT cross-domain sessions with domains specified in the resource routing definition (RRD) statements.

Note: The value for SECOPTS.OPERSEC in CNMSTYLE must have a value other than MINIMAL for span of control to be in effect.

NGMFADMN=NO|YES
Specifies whether operators using this profile are allowed to perform administrative functions for NetView management console. The NetView management console functions controlled by this operand are:
- Using the command profile editor
- Adjusting aggregation for individual resources
- Customizing views

NO Indicates that operators using the profile containing this statement do not have NetView management console administrative authority. NO is the default.

YES Indicates that operators using the profile containing this statement do have NetView management console administrative authority.

NGMFCMDS=YES|NO
Specifies whether operators using this profile are allowed access to NetView management console capabilities. NetView management console commands allowed by this operand are:
- Changing flags or notes on flags, for example, suspending a resource from the Resource Properties window.
- Changing non-SNA status
- Issuing a NETV390 command for a resource.

YES Indicates that operators using this profile, containing this statement, have NetView management console command capability. YES is the default.

NO Indicates that operators using this profile, containing this statement, do not have NetView management console command capability.

NGMFVSPN=span_level visible_objects restrict_view_info restrict_list_info
Defines the operator’s authority to display NetView management console views and resources within views. The NGMFVSPN attribute specifies whether each resource, each view name, or both will be checked in the NetView span table when an operator asks to display an NetView management console view. The attribute also specifies whether view lists and views will indicate that view names or resources have been excluded if the operator is not authorized to see an entire view or some resources in a view.
The NGMFVSPN attribute is coded as a character string. Use each of the 4 characters in the string to specify a different option for operator authorization to display NetView management console views and resources.

If there is a coding error in the `span_level`, `visible_objects`, `restrict_view_info`, or `restrict_list_info` setting, the default setting for that position is used, and an error message is sent to the authorized receiver. So, an error such as NGMFVSPN=VZNN would result in a system setting of NGMFVSPN=VNNN.

**span_level**

Defines what level of span checking, if any, is to be enabled when this operator requests NetView management console displays of views and resources.

- **N** None. Means the span table is not checked for operator authority. Because access is not checked, the operator can see all views and resources displayed by the NetView management console. N is the default.
- **V** Views. Means each view name is checked in the span table to see if the operator is authorized to display the view. This option avoids the overhead of span checking all resources in a view.
- **R** Resources. Means each resource is checked in the span table to see if the operator is authorized to display the resource. View names are not span checked, but every resource in a view is checked.
- **A** All. Means both view names and resources are checked in the span table to see if the operator is authorized to display them.

**visible_objects**

Specifies whether resources that are not in the operator’s span of control are visible as null nodes and links in views displayed to the operator. Null nodes and null links are placeholders that do not indicate the type of node or link or give any other information about a node or link except its placement in the network hierarchy.

This option applies only if you specify R or A for `span_level`.

- **N** Not visible. Any resources not in an operator’s span of control are not displayed in the operator’s views. N is the default.
- **Y** Visible. Any resources not in an operator’s span of control are displayed in the operator’s views as null nodes and links.

**restrict_view_info**

Specifies whether an indication should be given to the operator when objects not in the operator’s span of control are excluded from a view the operator requests or when an entire view cannot be shown to the operator because it is not in the operator’s span of control.

This option applies only if you specify R, V, or A for `span_level`.

- **N** Do not display restricted view information. The operator is not given an indication when resources are excluded from a view or a view is not displayed because the view or resources are not in the operator’s span of control. N is the default.
- **Y** Display restricted view information. When the operator does not have authority to see either an entire view or some resources in a requested view, the operator is given an indication that either the entire view or certain resources have been restricted from the operator’s display because they are not in the operator’s span of control.
restrict_list_info

Specifies whether an indication should be given to the operator when view names or resource names that are not in the operator’s span of control are excluded from lists. The types of lists include the list of views on the Graphic Monitor Details screen and the results of a Locate Resource, More Detail, or List Suspended Resources request.

This option applies only if you specify R, V, or A for span_level.

N  Do not display restricted list information. When view names are excluded from the view list or from Locate Resource responses because the operator is not authorized to see those views, the operator is not given any indication that view names have been excluded. N is the default.

Y  Display restricted list information. The operator is given an indication when view names are excluded from the view list or from a Locate Resource response because the views are not in the operator’s span of control.

Notes:

1. Use the NGMFVSPN attribute in AUTH statements only when you are using a NetView span table to define spans of control. If you do not use a NetView span table, and if you specify a CTL setting other than GLOBAL and any value other than N for the span_level position in the NGMFVSPN attribute, the operator is not able to display views and error messages are sent to the operator. If you specify CTL=GLOBAL and any value other than N for the span_level position in the NGMFVSPN attribute, the NGMFVSPN value for the operator is reset to NNNN, no span checking is done, and the operator is able to display all views and resources. An error message is also sent to the authorized receiver at operator logon.

2. If you are using RACF for RODM security, ensure that the NetView domain name is defined to RACF and has been permitted to a minimum of RODM security level 2.

The following table shows some span authorization options you might want to use and the NGMFVSPN setting for each level of span checking.

<table>
<thead>
<tr>
<th>Desired Span Checking Level</th>
<th>NGMFVSPN Setting</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>No span checking</td>
<td>NNNN</td>
<td>The NetView span table is not checked for operator authorization. The operator is shown all resources in whatever views the operator attempts to open. This option incurs no additional system overhead when the view is opened and should be used for all operators who have CTL=GLOBAL in their operator profiles. If you do not want to use the NGMFVSPN attribute now, but think you may want to use it later, you can code NGMFVSPN=NNNN as a placeholder.</td>
</tr>
<tr>
<td>Span checking for views only</td>
<td>VNNN</td>
<td>The NetView span table is checked for operator authorization to the view name before each view is displayed. If the operator is not authorized to display a requested view, the operator receives an error message.</td>
</tr>
<tr>
<td>Desired Span Checking Level</td>
<td>NGMFVSPN Setting</td>
<td>Results</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Span checking for resources only</td>
<td>RYNN</td>
<td>The NetView span table is checked for operator authorization to the resources in a view before each view is displayed. The view name itself is not checked, but every resource in the view is checked. If the operator is not authorized to see some of the resources in a requested view, those resources are shown as null nodes and links in the view that is displayed to the operator. If none of the resources in a requested view pass the span check for that operator, an error message is sent to the operator. This option incurs higher system overhead than view-level checking when the view is opened.</td>
</tr>
<tr>
<td>Span checking for all views and resources</td>
<td>ANYY</td>
<td>The NetView span table is checked for operator authorization to both the view name and every resource in the view before each view is displayed. If the operator is not authorized to display some of the resources in the view, those resources are excluded from the view before it is displayed to the operator. The excluded objects are not shown as null nodes and links, but the operator is given an indication that resources are excluded from the view. If either the view name or all of the resources in a requested view fail the span check for that operator, an indication is sent to the operator. This option incurs higher system overhead than both view-level checking and resource-level checking when the view is opened.</td>
</tr>
</tbody>
</table>

**Related Statements:** DOMAINS, ISPAN, OPERATOR, SECOPTS, PROFILE, PROFILEN, SPAN

**AUTOCMD**

The AUTOCMD statement loads the automation table. You can specify any number of automation tables. You can also use the AUTOCMD statement to specify a listing member name and marker value used by the AUTOMAN command. This statement is coded in CNMSTYLE.

The syntax for the AUTOCMD statement is:

```
AUTOCMD

| table_name, ORDER=sort_value, LIST=member_name, MARKER=value |
```

Where:

- **table_name**
  - Is the name of the automation table.

- **ORDER**
  - Specifies to load the automation table in the specified order.
  - **sort_value**
    - Indicates the order the tables are loaded by the EBCDIC value of the characters specified. You can also specify the following:
**AUTOCMD**

*FIRST*
Inserts the table as the first automation table.

*LAST*
Inserts the table as the last automation table.

**LIST**
Indicates to create an automation table listing.

*member_name*
Specifies the member in which the NetView program places the listing output it creates. You can specify *NONE* if you do not want a listing.

**MARKER**
Specifies to create a marker to be used with the AUTOMAN command.

*value*
Specifies the marker value.

**Usage Notes:**
- If AUTOCMD."table_name".ORDER is omitted, the specified automation table will not be loaded.
- AUTOCMD."table_name".MARKER is optional. The marker is used with the AUTOMAN command.
- You can also use the AUTOTBL command to load and test automation tables. Additionally, the automation table management command (AUTOMAN) enables you to work with individual or multiple automation tables through a full-screen panel interface. For more information, refer to the Tivoli NetView for z/OS Command Reference.

**AUTOTASK**

The AUTOTASK statement is used to start an automated operator and optionally associate an MVS console with this automated operator. This statement is coded in CNMSTYLE.

The syntax for the AUTOTASK statement is:

**AUTOTASK**

```plaintext
AUTOTASK.autotask_name.Console=console_name.InitCmd=command
```

**Where:**

*autotask_name*
Is the autotask name.

*console_name*
Specifies a 2–8 character MVS console name or ID to be associated with the autotask. You can also specify the following:

*ANY*
Assigns an autotask to respond to commands from any console not otherwise assigned to an autotask.

*MASTER*
Assigns an autotask to respond to commands from the current active master console.
AUTOTASK

*NONE*

Does not assign a console to be associated with the autotask.

command

Specifies the initial command to be used when the autotask is started, instead of the initial command from the task’s profile.

Usage Notes:

- The autotask does not process commands until CNMSTYLE completes.
- You can also use the AUTOTASK command to start an automated operator and optionally associate an MVS console with this automated operator. For more information, refer to the Tivoli NetView for z/OS Command Reference or the NetView online help.

auxInitCmd

The auxInitCmd statement defines a command or a command list to run automatically when the NetView program is started. This statement is coded in CNMSTYLE.

The syntax for the auxInitCmd statement is:

```plaintext
auxInitCmd.value=command
```

Where:

value

Specifies the EBCDIC value that determines the order in which the command is to run. Value can be up to 21 characters in length.

command

Specifies the command or command list to be run.

Usage Note: The commands specified on the auxInitCmd statements run before any commands for any autotask. All commands for autotasks, including both task initial clist and commands sent by EXCMD, are queued and held up. They are run only after all auxInitCmds have completed. Messages are also queued; they are not submitted to automation nor logged until all auxInitCmds have completed.

banner

The banner statement specifies customized identification information that will be displayed on the NetView 3270 logon panel and the command facility panel.

The syntax for the banner statement is:

```plaintext
banner = string
```

Where:

string

Specifies up to 24 characters of data that will be displayed on the 3270 logon
and command facility panels. Do not enclose the character string in quotation marks. If more than 24 characters are entered, the data is truncated. A single asterisk indicates that customized identification information will not be displayed.

Usage Note: After the NetView program is initialized, you can use the DEFAULTS command to dynamically change the value for the banner information. For more information on the DEFAULTS command, refer to the Tivoli NetView for z/OS Command Reference or the NetView online help.

C (Command List)

The C (Command List) statement enables you to assign up to 16 command lists to be displayed on the Status Monitor Detail panel.

Command lists can be written in REXX or the NetView command list language. Code this statement in DSICNM. Changes to DSICNM do not take effect until you stop and restart the $nv_id$VMT task (where $nv_id$ is the NetView domain identifier). If you change the member and the change results in additional extents being used, stop and restart the NetView program to implement the changes.

Code one C statement for each command list you define. You can code up to 16 C statements.

The syntax for the C statement is:

```
C cmdlistname comments
```

Where:

C  Must be in column 1.

`cmdlistname`
Is the name of the command list you want to display on the Status Monitor Detail panel. This value must begin in column 3 of the statement.

`comments`
Is unused space in the statement. Any comments you include with the statement must be in columns 11–80.

CCDEF

The CCDEF statement specifies a DSIPARM member from which command definitions are to be read. This statement is coded in CNMSTYLE.

The syntax for the CCDEF statement is:

```
CCDEF=member_name
```

Where:
CCDEF

member_name

Specifies the DSIPARM member name.

Usage Note: For information on correlated command definitions, refer to sample DSICCDEF.

CMDCLASS

The CMDCLASS statement is obsolete. It is used for migration purposes only. For more information, refer to the Tivoli NetView for z/OS Installation: Migration Guide.

Related Statements: KEYCLASS, OPCLASS, VALCLASS

CMDLINE

The CMDLINE statement enables you to change the color of the command line that is displayed on the command facility panel. The sample member containing the CMDLINE statement is CNMSCNFT. You can code the CMDLINE statement only once.

The syntax for the CMDLINE statement is:

CMDLINE

| colorf | attribute | ON colorb |

Where:

colorf

Defines foreground color for the command line. The foreground color must be specified before the background color.

attribute

Defines alarm, intensity, and highlight attributes for the command line. You can specify attributes only once for the CMDLINE statement.

ON

Makes the color following ON apply to the background of the command line. This is a required keyword if you specify only a background color.

colorb

Defines background color for the command line.

Usage Notes:

• You can also create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

• The NetView-supplied defaults for the CMDLINE statement are:
  – Green foreground
  – Black background
  – Normal highlighting
  – Low intensity
  – Alarm off

• The color operands are:
The CMDMDL statement defines commands and command lists to the NetView program.

For commands, the CMDMDL shows the following:

- The command name
- The module that processes the command
- The command type
- When the command module is loaded
- Whether authority verification should be performed

Use a CMDMDL statement for command lists for the following:

- To define synonyms for the command list
- To explicitly check or bypass authority checking for a particular command or command list

Also, use a CMDMDL statement for commands when you want to perform authorization checking on non-NetView commands. For example, you may want to use DSIEX19 to check the commands passed to a service point using the RUNCMD command. To do this, use MOD=DSISPCMD on the CMDMDL statement.

The CMDMDL statement is used in DSICMD. Stop and restart the NetView program to implement the changes.

The syntax for the CMDMDL statement is:
Where:

commandname

Is the name of the command or command list. The name must begin in column 1. Do not use the percent (%) character in the command name, because it is reserved for the NetView program.

MOD=modulename

Shows the name of the module that processes the command. For command lists, use MOD=DSICCP.

Note: When you define a user-written command procedure, be sure to specify a unique module name on the MOD operand. Do not use a name that the system might recognize as a command, because the NetView program attempts to execute that command instead of the user-written command procedure.

TYPE=B|BP|D|H|I|P|R|RD|RP

Shows the command type.

Note: Do not define regular command procedures with TYPE=I or TYPE=B.

B Indicates a command that can run as a regular or an immediate command. If an operator enters the command at a terminal, NetView treats it as an immediate command. If the command is in a command list, the NetView program treats it as a regular command.

BP Indicates a both or stage command.

D Indicates a data services command. Data services command procedures run under a data services task (DST). Because these commands are internal, operators cannot enter them at their
terminals. Some internal commands that support the session and hardware monitors are TYPE=D.

H Type H is the same as type R except that type H commands are queued at high priority regardless of the setting of CMD priority of the DEFAULTS and OVERRIDE commands.

I Indicates an immediate command. An immediate command interrupts a regular command. Immediate commands are usually screen-control or error-recovery commands. When an immediate command is running, the keyboard is locked, and you cannot enter other commands. Immediate commands are not allowed in command lists or as initial commands. Also, you cannot use immediate commands with the PPT operand of the AT, EVERY, and AFTER commands.

P Indicates a stage command within a PIPE command. A command of type P controls the activity of a stage command within a pipeline.

R Indicates a regular command. Only one regular command runs at a time for an operator. If an operator enters a regular command while another regular command is running, the second command is held by the NetView program and runs after the first command is finished. Regular commands are allowed in command lists. R is the default.

RD Indicates a regular or data services command.

RP Indicates a regular or stage command.

RES=N|Y Specifies whether the command module is loaded when the NetView program is initialized. By specifying that only frequently used modules are loaded at initialization, you increase the amount of storage available to the NetView program. If you load all command modules at initialization, performance can be degraded because less operating storage is available.

N Indicates the command module is not loaded until the command is run, and the storage that is used is freed upon completion.

Y Indicates the command module is loaded at NetView program initialization and remains resident in storage. Y is the default.

If you have duplicate CMDMDL statements that have conflicting residency attributes, message DSI629I is issued and RES=Y is assumed. If the CMDMDL statement is TYPE=I or TYPE=B and RES=N, message DSI813A is issued, RES=Y is assumed, and initialization continues. If you code the following commands as RES=N, the NetView program automatically makes them RES=Y:

- All VTAM commands
- DSICCP (command lists command processor)
- BGNSESS, DSICYCLE, DSICYCTR, DSICYORM, DSICYOFPM, DSICYOLRP, DSICYOMRP, DSICYORP, DSICYOTP, DSICYORP, LOGOFF, REPLY, START, STOP

ECHO=N|Y Controls whether a command is echoed to the screen after an operator enters it. For commands in NetView command list language command lists, the setting for &CONTROL determines the echo status. TYPE=I and TYPE=B commands are not echoed after they are entered from a terminal.

N Means an entered command is not echoed.
CMDMDL

Y Means an entered command is echoed. Y is the default.

Note: Commands that are echoed to the screen are subject to NetView automation if they match conditions in the automation table.

PARSE=N|Y
Specifies whether to parse the message buffer. This operand is ignored for command list invocations.

N Indicates that the message buffer is not parsed. PARSE=N improves command processor performance for user-written command procedures that do not use the parse buffer PDB.

Y Indicates to parse the message buffer. Y is the default. For information on how to write command procedures, refer to Tivoli NetView for z/OS Customization: Using Assembler and to Tivoli NetView for z/OS Customization: Using PL/I and C.

SEC=BY|CH|DE
Specifies whether to perform security verification on the command. All commands are authority-checked by default. Specify SEC on your CMDMDL statement only if you want to explicitly bypass authority checking or to explicitly check the command regardless of the setting of AUTOSEC on the DEFAULTS command.

BY BYPASS specifies that the NetView program will not perform authority verification for the command. This overrides AUTOSEC=CHECK as specified on the DEFAULTS command.

If you specify SEC=BY and command authorization table statements or SAF statements for the same command, the command security statements are ignored.

Specifying SEC=BY to bypass command authorization checking for commands that are frequently used and present no security risk can result in an overall savings in NetView processor utilization.

CH CHECK - specifies that the NetView program will always perform authority verification when this command is issued. This overrides AUTOSEC=BYPASS as specified on the DEFAULTS command.

You should use SEC=CH to restrict access to commands that when misused could have negative effects in your environment. This statement causes an authorization check and prevents a task from issuing an unauthorized command, even if the command originated from the automation table and AUTOSEC is set to BYPASS.

DE DEFER specifies that the NetView program will conditionally perform authority verification for the command.

If the command is issued from the automation table, authority checking is performed depending on the value assigned to AUTOSEC by the DEFAULTS command. If AUTOSEC=CHECK, the command is authority-checked. If AUTOSEC=BYPASS, the command is not checked.

IGNRLSUP=*|number
Specifies whether or not to ignore suppression of logging the command when the suppression characters are coded on a specific command.

* Logs the entire command string.

number The value 0 indicates that suppression characters should be honored.
Values 1 through 250 indicate that the first \( n \) parsed tokens (character strings delimited by blank, comma, period or equal sign) of the command string should be logged.

**Note:** Specifying IGNRLSUP=1-250 with PARSE=N for the same command definition is considered an error.

The command verb is the first parsed token. For example IGNRLSUP=1 logs only the command verb.

**Notes:**

1. The IGNRLSUP value is honored only if the command is not suppressed for any other reason. For example, if the command is run from a command list and &CONTROL is set to ERR, then the command is suppressed.
2. The IGNRLSUP value coded on the ADDCMD command or CMDMDL statement takes precedence over any value coded on the DEFAULTS command. If IGNRLSUP is not coded on the DEFAULTS or ADDCMD commands or the CMDMDL statement for the command, then the suppression characters are honored for that command.
3. The IGNRLSUP value is ignored when processing commands in a command list.

**Usage Notes:**

- Use the COMNTESC statement just before a CMDMDL statement if you need to define a command that begins with, or is, an asterisk (*). The COMNTESC statement forces the NetView program to treat the next definition statement as if the * in column 1 does not denote a comment.

---

**Product-sensitive programming interface**

- You can use installation exit DSIEX19 to perform authorization checking for non-NetView commands sent to a service point using the RUNCMD command. If you want to use DSIKVS (assembler) or CNMSCOP (high-level language) to perform authorization checking on keywords and values, you must include a CMDMDL statement in DSICMD for non-NetView commands.

---

**End of Product-sensitive programming interface**

**Related Statements:** CMDSYN, COMNTESC, SECOPTS, PARMSYN

---

**CMDRCVR**

The CMDRCVR statement enables you to define the NetView program-to-program interface command-receiver queue used to send commands from RODM methods for execution in the NetView program. If you specify more than one CMDRCVR statement, the NetView program uses only the first statement.

The syntax for the CMDRCVR statement is:

**CMDRCVR**

```
CMDRCVR ID=receiverid
```

```
CMDRCVR ID=DSIQTSK
```
CMDRCVR

Where:

ID=DSIQTSK | receiverid

Specifies the receiver ID used by senders to send commands over the
program-to-program interface to the command receiver. RODM methods must
specify this name when running the EKGSPPI object-independent method to
send commands to this NetView program for execution.

DSIQTSK

Indicates the name of the DSIQTSK task in CNMSTASK. This is the
default.

receiverid

Specifies the receiver ID of the command receiver. The length of this
value cannot exceed 8 characters.

CMDSYN

The CMDSYN statement enables you to define additional names for commands or
command lists. Code this statement in DSICMD, then stop and restart the NetView
program to implement the changes.

Multiple CMDSYN and PARMSSYN statements can follow each CMDMDL
statement. The CMDSYN statements must follow all of the PARMSSYN statements
for a CMDMDL; otherwise, the operand synonyms for that CMDMDL are ignored.

The syntax for the CMDSYN statement is:

CMDSYN

Where:

label

Indicates the optional label for the CMDSYN statement. This label identifies
the statement in any related error messages.

synonym

Indicates the new name for the CMDMDL statement that precedes this
CMDSYN statement. The operator can use either the original command name
or the new synonym.

Usage Notes:

- Do not use the following as command synonyms:
  - A comma or a period
  - A name that begins with a percent sign (%)
  - A name that is a VTAM command
  - Another NetView command
  - A command in an application program that runs with the NetView
    program
- NetView help is keyed to command names. If you create a synonym, you
can create help for that synonym or add the synonym to the helpmap
sample (CNMS1048). Refer to the [Tivoli NetView for z/OS Customization Guide](#) for more information on writing help panels.

- The synonyms might not work as expected, depending on the NetView component you are using. For example, you can assign the T synonym to the TARA command using the CMDSYN statement. However, if you are in the hardware monitor component, T executes the TITLE command instead of the TARA command.
- The system console operator must always use the original command name.

See the DSICMD sample for the Tivoli-defined NetView synonyms and then customize DSICMD for your NetView program use.

**Related Statements:** CMDMDL, PARMSYN

---

**CNMI**

The CNMI statement defines whether this NetView program owns the Communications Network Interface (CNMI). This statement is coded in CNMSTYLE.

The syntax for the CNMI statement is:

```
CNMI
```

Where:

- **No**
  - Indicates that another NetView program will have the CNMI. Specify No if you are configuring a secondary NetView program.
- **Yes**
  - Indicates that this NetView program will have the CNMI. This is the default.

**Usage Note:** Refer to [Tivoli NetView for z/OS Installation: Configuring Additional Components](#) for more information on configuring the multiple NetView programs.

---

**COLUMNHEAD**

The COLUMNHEAD statement enables you to change the color of the column heading line that appears on the command facility panel. The column heading line is displayed at the top of the output area to define which prefixes are in use. The sample member containing the COLUMNHEAD statement is CNMSCNFT. Code the COLUMNHEAD statement only once.

The syntax for the COLUMNHEAD statement is:

```
COLUMNHEAD
```

---
COLUMNHEAD

Where:

colorf
Defines foreground color for the column heading line. Specify the foreground color before the background color.

attribute
Defines alarm, intensity, and highlight attributes for the column heading line. You can specify attributes only once for the COLUMNHEAD statement.

ON
Makes the color following ON apply to the background of the column heading line. This is a required keyword if you specify only a background color.

colorb
Defines background color for the column heading line.

NONE
Indicates that labels should not be displayed. NONE is the default.

Usage Notes:

- You can create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The color operands are:
  BLACK  The color black
  BLUE    The color blue
  RED     The color red
  PINK    The color pink
  GREEN   The color green
  TURQ    The color turquoise
  YELLOW  The color yellow
  WHITE   The color white
- The highlight operands are:
  BLINK   The characters blink.
  REV     The characters change to reverse video.
  UND     The characters are underlined.
  NRM     Normal attributes are used.
- The intensity operands are:
  HIGH    The characters have high intensity.
  LOW     The characters have low intensity.
- The alarm operand is:
  BEEP    An audible alarm sounds.

Related Statements: ACTION, CMDLINE, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

COMMON

The COMMON statement sets common global variables. This statement is coded in CNMSTYLE.

The syntax for the COMMON statement is:
COMMON

```plaintext
 COMMON. variable_name=value
```

Where:

- `variable_name` specifies the name of the common global variable.
- `value` specifies the value of the common global variable. This value cannot exceed 256 characters. You can use a system symbolic. The value can be continued on subsequent lines (with a leading blank).

Usage Notes:
- The variables are set before any autotasks are started and before automation is enabled.
- You can also use the GLOBALV command to set a common global variable from a REXX or NetView command list. For more information, refer to the [Tivoli NetView for z/OS Command Reference](http://www.ibm.com/doc) or the NetView online help.

COMNTESC

The COMNTESC statement causes the next statement in DSICMD to be processed even if the first character on that statement is an asterisk. COMNTESC applies only to the following single statement. The COMNTESC statement can only be used in DSICMD definitions or in the %INCLUDE files within a DSICMD definition hierarchy.

The syntax for the COMNTESC statement is:

```plaintext
 COMNTESC
```

Usage Notes:
- Place the COMNTESC statement directly before the CMDMDL statement. Do not use COMNTESC as the last statement within a %INCLUDE member. Instead, place the COMNTESC statement in the same member as the statement that it affects. The NetView program does not check for this condition.
- The following example shows how to use the COMNTESC statement:

```plaintext
 DSIOUSNDM CMDMDL MOD=DSIOUSNDM,PARSE=Y,TYPE=RD,RES=Y
    CMDSYN RMTCMD
    OPERID COMNTESC
    *OTHER
```

Related Statements: CMDMDL
COS

The COS statement defines equivalent names for duplicate class-of-service (COS) names in an interconnected network. Code this statement in the members specified in the ALIASMEM statement. The sample member supplied with the NetView program is DSIALTAB.

The syntax for the COS statement is:

```
COS
```

Where:

- **label**
  - Indicates the optional label for the COS statement. This label identifies the statement in any related error messages.

- **targname**
  - Indicates the COS name that applies to the target network. This name is also called the equivalent name because it becomes equivalent to the COS name in the origin network.

- **targnet**
  - Indicates the 1-8 character name of the network where the COS is known by its equivalent name.

- **origname**
  - Indicates the name by which the COS is known in the origin network.

**Related Statements:** ALIASMEM, ORIGNET

---

DB2SEC

You can use the DB2SEC statement in CNMSTYLE to:

- Load CAF interfaces. In this case, you do not have operator level security, but can access multiple DB2 subsystems on your system.
- Load RRS interfaces. This allows for operator level security checking and access to multiple DB2 subsystems on your system.

Once the CAF or RRS interfaces are loaded, tasks can access DB2 directly without needing the DSIDB2MT task. For tasks that access the DB2 subsystem that DSIDB2MT accesses, DSIDB2MT must still be active. Starting the DSIDB2MT task makes sure that SQL requests that do not specify which DB2 to access, will always run the same DB2. The SSID ssidname!* parameter on the SQL stage defines whether a specific or the default DB2 is accessed. Once an SQL stage defines a subsystem to access, that subsystem remains in effect for that task until you reset it by using another SQL stage.

The syntax for the DB2SEC statement is:
DB2SEC

Where:

*NONE*
   Specifies not to use DB2. This is the initial setting in CNMSTYLE.

CAF
   Specifies to use the DB2 interfaces that give all tasks the same security (the
   address space user identity).

RRS
   Specifies to use the DB2 interfaces that provide security on a per-task basis.

DBFULL

The DBFULL statement enables you to set the maximum number of times that
message BNJ022I is issued when the hardware monitor database is full. When the
limit you specify is exceeded, the NetView program issues message BNJ183I. Once
message BNJ183I is issued, no more messages are issued until one of the following
situations occurs:

• You stop the BNJDSEVR task.
• You issue the SWITCH command for BNJDSEVR.
• A PURGE command completes.

If you do not code the DBFULL statement, message BNJ022I is always issued. If
you set DBFULL to zero, neither message BNJ022I nor message BNJ183I is issued.

Code this statement in the member specified by the MEM keyword of the
BNJDSEVR task statement. The sample member supplied with the NetView
program is BNJMBDST.

To implement the changes, stop and restart the BNJDSEVR task. If the changes
result in additional extents being used, stop and restart the NetView program.

The syntax for the DBFULL statement is:

DBFULL

Where:

nnnnn
   Indicates the number of times that you see message BNJ022I. When nnnnn is
   exceeded, message BNJ183I is issued.

DEFAULTS

The DEFAULTS statement sets initial values for NetView-wide defaults. The
DEFAULTS definition statement is defined in CNMSTYLE.
After the NetView program is initialized, you can use the DEFAULTS command to dynamically change values. Refer to the NetView online help for more information about the DEFAULTS command.

The syntax for the DEFAULTS statement is:

```
DEFAULTS
```

```
keyword = value
```

Where:

- **keyword**
  Specifies the DEFAULTS command keyword to be initialized. For more information on the keywords you can specify, refer to the DEFAULTS command in the NetView online help.

- **value**
  Specifies the DEFAULTS command keyword value to be used. For more information on the values you can specify, refer to the DEFAULTS command in the NetView online help.

Usage Notes:
- The HCYLOG, NETLOG, SYSLOG initial settings take effect after the corresponding logging task becomes active.
- Use the Banner statement in CNMSTYLE rather than the DEFAULTS statement to initialize the banner.

---

The DEFENTPT statement is an optional statement used in conjunction with the DEFFOCPT statement to specify whether a NetView node is initialized as an operations management focal point or entry point. Code DEFENTPT in the DSI6INIT member. DSI6INIT is the data services task (DST) initialization member for DSI6DST, which is the management services (MS) transport DST.

The syntax for the DEFENTPT statement is:

```
DEFENTPT
```

```
label DEFENTPT EPONLY=NO
EPONLY=YES
```

Where:

- **label**
  Indicates the optional label for the DEFENTPT statement. This label identifies the statement in any related error messages.

- **EPONLY**
  Specifies whether a node is an entry point only.
NO
Specifies that this NetView node is an operations management focal point.
If no DEFFOCPT and no DEFENTPT statements are defined, NO is the
default.

YES
Specifies that this NetView node is capable of being an entry point for
operations management data. This is the only allowable value if a valid
DEFFOCPT statement is defined for operations management, or if a focal
point has been restored from the Save/Restore database for operations
management.

Usage Note: If you specify more than one DEFENTPT statement, the later one
overrides the previous one. This occurs even if the later statement has a syntax
error.

Related Statements: DEFFOCPT

DEFFOCPT

The DEFFOCPT statement defines focal points for the type (category) of data
specified. If the focal points being defined are for LUC alert forwarding, place the
DEFFOCPT statements in the DSICRTTD member. If the focal points are for any
other type of data, including LU 6.2 alert forwarding, place the DEFFOCPT
statements in the DSi6INIT member.

To implement changes to DSICRTTD stop and restart the DSICRTR task. Changes
to DSi6INIT take effect when either of the following occurs:
• An operator at the entry point issues the FOCALPT ACQUIRE DEFFOCPT
  command.
• The DSi6DST task is restarted.

If the changes result in additional extents being used in the DSIPARM, stop and
restart the NetView program.

There are two syntaxes for the DEFFOCPT statement. The syntax you use depends
on where the DEFFOCPT statement is located. Refer to "Centralizing Operations"
in the Tivoli NetView for z/OS Installation: Configuring Additional Components for
eamples of how to code the statement. Refer to Tivoli NetView for z/OS Automation
Guid for a general discussion of NetView focal point and entry point support.

DEFFOCPT Statement in DSICRTTD

Use the following syntax if the DEFFOCPT statement resides in DSICRTTD. If you
code more than one DEFFOCPT statement, the last one coded is used.

Note: There are also DEFFOCPT statements residing in DSi6INIT.

DEFFOCPT
**DEFFOCPT**

```
label
```

**Where:**

*label*

Indicates the optional label for the DEFFOCPT statement. This label identifies the statement in any related error messages.

**TYPE=ALERT**

Defines the focal point category. You need to specify the TYPE keyword because no default is assigned.

**ALERT**

Indicates that alert data for a given NetView hardware monitor is forwarded over the LUC to the focal point.

Focal point alert support makes centralized problem determination easier by providing centralized alert notification.

**PRIMARY=nodename**

Indicates the name of the LUC task of the primary focal point. LUC tasks are named `xxxx`LUC, where `xxxx` is the NetView program identifier of the NetView program in which the LUC task runs.

**BACKUP=nodename2**

Indicates the name of the LUC task of an optional backup focal point. LUC tasks are named `xxxx`LUC, where `xxxx` is the NetView program identifier of the NetView program in which the LUC task runs. When you define a backup focal point, do not specify your local domain as its own backup focal point.

**Note:** If a primary focal point is not active and a backup focal point is defined, the NetView program sends the alert to the defined backup focal point.

**DEFFOCPT Statement in DSI6INIT**

Use the following syntax if the DEFFOCPT statements reside in DSI6INIT. You can specify multiple DEFFOCPT statements for a category, enabling you to specify the primary focal point and up to eight backup focal points. You can specify only one primary focal point for a category. If you specify more than one primary focal point for a category, the last primary focal point read from DSI6INIT is used. If you specify more than eight backup focal points, the first eight are used.

**Note:** There is also a DEFFOCPT statement residing in DSICRTTD.

```
DEFFOCPT2
```

**Where:**
DEFFOCPT

label
Indicates the optional label for the DEFFOCPT statement. This label identifies the statement in any related error messages.

TYPE=category
Defines the focal point category. This operand enables you to specify the initial primary and backup focal point settings for the specified category. You must specify the TYPE keyword because no default is assigned.

Valid focal point categories include ALERT, OPS_MGMT, and user-defined categories.

PRIMARY=netid.nau
Indicates the name of the network and LU or VTAM CP name in which the focal point resides. For the NetView program, the LU name is the NetView domain name. The PRIMARY keyword is required on at least one DEFFOCPT statement for a particular category. netid is optional. If you either specify an asterisk (*) or do not specify a network name for the netid, VTAM determines the network name of the LU.

You can define only one primary focal point for each category. If more than one is found, the last primary focal point read from DSI6INIT is used. If you do not define a primary focal point for a category, all DEFFOCPT statements for the category are ignored.

BACKUP=netid.nau2
Indicates the name of the network and LU or VTAM CP name in which the backup focal point resides. The BACKUP keyword is optional. netid is optional. When you define a backup focal point, do not specify your local domain as its own backup focal point. If you specify an asterisk (*) or do not specify a network name for the netid, VTAM determines the network name of the LU.

If you lose communication with your primary focal point, and a backup focal point is defined, the NetView program tries to acquire the backup focal point. The order in which backup focal points are acquired is determined by the sequence of DEFFOCPT statements within DSI6INIT. If the backup focal point is acquired, it becomes the current focal point. A timer is then activated to try to reacquire the primary focal point. For more information about defining, acquiring, and changing focal points, refer to Tivoli NetView for z/OS Automation Guide.

Refer to the REACQPRI keyword on the DEFAULTS command in the NetView online help for more information about specifying the time interval for reacquiring the primary focal point.

OVERRIDE
Specifies that all DEFFOCPT statements be used at initialization regardless of whether any focal point details for this category are found in the Save/Restore database. This keyword must appear on only one DEFFOCPT statement for a particular category and must be specified with either the PRIMARY or BACKUP keyword for that category. If you do not specify OVERRIDE, current focal point details from the Save/Restore database are used unless a DEFFOCPT statement for this category has been modified since the last time the DSI6DST task was started. In that case, the NetView program uses the focal point names defined in the DEFFOCPT statement.

Usage Notes:
• If two nodes in two different networks have the same LU name, the one that VTAM finds can vary depending on the configuration of nodes that are active at any given time.
DEFFOCPT

- If you specify both a qualified and an unqualified network name for a category with the same NAU, the qualified name is used.
- You can use the VTAM CP name only for the focal points residing in Version 2 Release 4 or a later release NetView program running under Version 4 Release 1 or a later release VTAM program.
- If the focal point name at a given node represents a NetView program, consider whether that NetView program uses VTAMCP .USE = YES or VTAMCP .USE = NO. The VTAMCP statement is coded in CNMSTYLE. If YES is specified, the focal point name should be that VTAM program’s CP name. If NO is specified, the NetView domain name should be used.

Related Statements: DEFENTPT, NPDA.ALERTFWD

DOMAIN

The DOMAIN statement specifies the NetView domain name as a global variable. This statement is coded in CNMSTYLE.

The syntax for the DOMAIN statement is:

```
DOMAIN
```

Where:

`domainid`

Indicates a 1–5 character domain name for this instance of the NetView program. The NetView-supplied default is CNM01 (C&NV2I.01).

Usage Notes:

- If you specified the NetView domain ID or password in CNMPROC (CNMSJ009), the DOMAIN statement in CNMSTYLE or the ACBpassword statement in CNMSTPWD do not have to be coded. If they are coded, they are ignored unless the parameters passed by CNMPROC are null. If the domain password is not specified in CNMPROC (CNMSJ009) or in CNMSTYLE, the DOMAIN name becomes the password.

DOMAINS

The DOMAINS statement enables the setup of NNT cross-domain communication for operators with specific or general control. This statement shows which NNT cross-domain sessions this operator can start. You code this statement in the member specified by the PROFILEN statement associated with the operator. See "OPERATOR" on page 150 and "PROFILEN" on page 161 to determine how a PROFILEN statement is associated with an operator.

The syntax for the DOMAINS statement is:
DOMAINS

Where:

label
  Indicates the optional label for the DOMAINS statement. This label identifies
  the statement in any related error messages.

domainid [,...]
  Indicates a 1–5 character identifier of another NetView where this operator can
  start an NNT cross-domain session.

Usage Notes:

• Conversations with distributed autotasks started with the RMTCMD command
  are not considered NNT cross-domain sessions and are not affected by the
  DOMAINS statement.

• You can use the DOMAINS statement as many times as necessary to define all
  the required domains. The DOMAINS statement is ignored if you specify either
  AUTH CTL=GLOBAL or SECOPTS.OPERSEC=MINIMAL.

Related Statements: AUTH, OPERATOR, SECOPTS, PROFILEN

DSTINIT

The DSTINIT statement defines the initialization operands required by optional
data services tasks (DSTs).

Code this statement in members specified by the MEM keyword of tasks whose
MOD keyword value is DSIZDST. For example, task DSIAL2WS is coded in
CNMSTYLE as follows:

```
TASK.DSIAL2WS.MOD=DSIZDST
TASK.DSIAL2WS.MEM=DSIBKMEM
TASK.DSIAL2WS.PRI=8
TASK.DSIAL2WS.INIT=Yes
```

Because MEM=DSIBKMEM, the DSTINIT statement for task DSIAL2WS is coded
in member DSIBKMEM:

```
DSTINIT FUNCT=OTHER
DSTINIT DSRBO=1
DSTINIT XITDI=DSIBKINT
```

To implement the changes to the member containing the DSTINIT statements stop
and restart the associated tasks. If the changes result in additional extents being
used, stop and restart the NetView program.

The syntax for the DSTINIT statement is:

```
DSTINIT
```
DSTINIT

Where:

**label**

Indicates the optional label for the DSTINIT statement. This label identifies the statement in any related error messages.

**Note:** If a label is not present, do not code the DSTINIT statement beginning in column 1 because this causes the DSTINIT statement to be interpreted as a label.
FUNCT=BOTH|CNMI|OTHER|VSAM

Specifies the task function. Do not code multiple DSTINIT statements in the same member with conflicting FUNCT definitions.

**BOTH**
Specifies that the task is to send CNM data to and receive CNM data from the network using VSAM. BOTH is the default.

**CNMI**
Specifies that this task sends CNM data to and receives CNM data from the network only. It is not necessary to specify UNSOL unless unsolicited CNM data is to be processed by the DST. You do not need to specify a primary data set.

**OTHER**
Specifies that this task does not use VSAM or CNM services. For example, this operand is used for the VTAM LU task or to define a sequential log task. If you specify DSRBU, it is ignored and set to zero. This operand does not preclude the task from opening an access control block (ACB); a VTAM LU task (DSIAMLUT) does.

**VSAM**
Specifies that this task uses VSAM services. Code the UNSOL and DSRBU parameters for the AAUTSKLP and BNJDSERV tasks. They receive unsolicited data from DSICRTR.

**PBSDN=name**
Indicates the 1–8 character DD name of the primary data set to be used by NetView sequential log services. Enter this name on a DD statement when you start the NetView program, or anytime before starting the sequential log tasks through the NetView ALLOCATE command. Specify this keyword only once in each sequence of DSTINIT statements if you use NetView sequential log services. The PBSDN keyword is required for sequential logging and causes the NetView program to define the task as a sequential task.

**PDDNM=name**
Indicates the 1–8 character DD name of the primary data set to be used by VSAM services. Enter this name on a DD statement when you start the NetView program. Specify PDDNM at least once in each sequence of DSTINIT statements if you use VSAM.

**UNSOL=name**
Indicates the 1–8 character verb that is the label of the CMDMDL statement for the data services command processor (DSCP) that gains control when unsolicited CNM data is received from the network or passed to the DST by the CNM router subtask (DSICRTR).

**PPASS=password**
Indicates a 1–8 character VSAM password to be used when the primary data set ACB is opened. This password must match the UPDPW operand in the primary VSAM database allocation.

**SBSDN=name**
Indicates the 1–8 character DD or FILEDEF name of the secondary data set to be used by NetView sequential log services. Enter this name on a DD or FILEDEF statement when you start the NetView program, or anytime before starting the sequential log tasks through the NetView ALLOCATE command. Specify this keyword only once in each sequence of DSTINIT statements if you use the NetView sequential log services. The SBSDN keyword is not required for sequential logging.
DSTINIT

SDDNM=\textit{name}
Indicates the 1–8 character DD or FILEDEF name of the secondary data set to be used by VSAM services. Define this name on a DD statement when you start the NetView program.

SPASS=\textit{password}
Indicates the 1–8 character password to be used when the secondary data set ACB is opened. This password must match the UPDPW operand in the secondary VSAM database allocation.

DSRBU=5\textit{nnn}
Is a decimal number from 0–999 that specifies the number of data services request blocks (DSRBs) to pre-allocate for processing unsolicited problem determination request units (RUs). The default value is 5.

DSRBO=3\textit{nnn}
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 3, except for NPDA task BNJDSERV. For NPDA task BNJDSERV, the default coded in CNMSTYLE is 5.

The following operands that begin with XIT are 1- to 8-character load module names for user-written exit routines for this DST. You can code up to 10 module names for each exit. However, if the number of module names needed exceeds the record length, restart the XIT statement on the next record. For example, assume that you code:

\texttt{DSTINIT XITCI=MODNAME1, XITCI=MODNAME2, XITCI=MODNAME3}

If you want to add MODNAME4 and MODNAME5, code the next record as follows:

\texttt{DSTINIT XITCI=MODNAME4, XITCI=MODNAME5}

These modules receive control in the order they are coded in the DST.

XITBN=\textit{modname[,...]}
Receives control when a sequential log data set is opened for output processing only (not for a resume). This exit allows you to put an initialization record into the file.

XITBO=\textit{modname[,...]}
Receives control before each NetView sequential log services record is blocked for output. XITBO can use only the service facilities available to a DST defined as FUNCT=OTHER, excluding the DSIWLS macro.

XITDI=\textit{modname[,...]}
Receives control as each DST initialization operand is read. This exit routine processes statements following this statement only. DSTINIT XITDI must precede any statement that is to be processed by the DST initialization exit.

XITCM=\textit{modname[,...]}
Receives control when you issue a remote command through command support of NetView management console. This exit routine receives a command to be issued and a domain to which the command is to be routed. This exit gives you an opportunity to specify the domain to which the command is routed. This exit overrides the RMTCMD parameter of OPERID=\texttt{*}.
XITCO=modname[,...]
Receives control for the CNM interface output processing before the request is passed to VTAM.

XITCI=modname[,...]
Receives control for the CNM interface input processing before the data received is passed to the DSCP. Specify this operand under the DSICRTR subtask for access to unsolicited CNM data prior to NetView program internal routing.

XITVN=modname[,...]
Receives control when an empty VSAM data set is opened for processing. This exit allows you to put an initialization record into the file. If you do not specify XITVN, access method services (AMS) are used to start the data set.

XITVI=modname[,...]
Receives control on input from the VSAM data set before the input record is passed to the data services command processor (DSCP).

XITVO=modname[,...]
Receives control before the output of a record to the VSAM data set.

XITXL=modname[,...]
Receives control before the data is logged to an external log. If SMF (MVS only) is not used, this exit can perform the necessary logging. XITXL receives control by coding the DSIWLS macro with the EXTLOG keyword.

MACRF=NSR|DFR|LSR
Specifies the local shared resource (LSR) options that meet the requirements of your installation. The default value is NSR, except for NPDA task BNJDSERV. For NPDA task BNJDSERV, the default coded in CNMSTYLE is LSR.

NSR
Indicates that the data set does not use shared resources.

DFR
Extends LSR to defer writing of records. The deferred write (DFR) option defers the writing of a record until the NetView program forces it out because buffer space is needed for a read. This further reduces I/O by minimizing writes.

If DFR is coded, do not use the MVS operator CANCEL command or the MVS STOP FORCE command to end the NetView task. Issuing these commands can damage the NetView databases by preventing the final writing of records for which output was deferred. To end the NetView task, use the NetView CLOSE or CLOSE IMMED commands.

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.

You can use DFR and LSR with the VSAM local shared resource (LSR) option. LSR is explained in OS/VS Virtual Storage Access Method (VSAM): Options for Advanced Applications. The possible values indicate the following:

DFR
Buffer pool scanning and deferred writes.

LSR
Buffer pool scanning on input only.
PERSIST=YES | NO
Defines the default for whether all sessions between this NetView program and a remote NetView program are persistent. Code this keyword in the member identified by the MEM keyword of the TASK statement that has a TSKID of xxxx:LUC.

YES
Indicates that the sessions are persistent and remain active regardless of the time elapsed between conversations between this NetView program and a remote NetView program. YES is the default value of the PERSIST keyword.

NO
Indicates that the sessions are nonpersistent and end if the time elapsed between conversations between this NetView program and a remote NetView program reaches the nonpersistent sessions timeout interval in DSICTMOD.

The PERSIST operand of the CNMTARG statement can override this default for specific LUs.

RETRY=YES | NO
Defines the default for whether an attempt to allocate a conversation between this NetView program and all the domains specified by NLDM.CDRMDEF statements in CNMSTYLE is made every 10 minutes if the first attempt fails. The RETRY setting is controlled by the NLDM.RETRY statement in CNMSTYLE.

YES
Indicates the session monitor attempts to establish an initial conversation with the other domains once every 10 minutes. YES is the default.

NO
Indicates the session monitor attempts to establish an initial conversation with the other domains only once.

The RETRY specification on the NLDM.CDRMDEF statement can override this default for specific domains.

Usage Notes:
• If the task is an LUC, AM/LU, DSITRACE, or DSILog, the NetView program sets DSRBU to zero.
• If the task is an LUC, AM/LU, DSITRACE, or DSILog, the NetView program sets DSRBO to 1.
• If the task is DSICRTR (router), the default in the DSICRTDD sample is 5. However, you may need to increase this value as indicated in the sample.
• If the task is CNMTAMEL, set DSRBO to 1.
• Refer to “Centralizing Operations” in the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information on calculating this number.
• If MACRF=DFR is defined, do not use the MVS operator RESET command to stop the NetView program because records that are deferred might not be logged, which corrupts the database. Use the NetView CLOSE or CLOSE IMMED command instead.
• If you are not using the LSR option, do not code this statement. Refer to "Centralizing Operations" in the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information about using the VSAM LSR option with the NetView program.

• The NetView program supports the LSR or DFR facility for the network log as well as the standard VSAM support for CNM applications. Do not use LSR or DFR for the DSILog task if you want to browse the network log.

• You can use PERSIST=YES|NO in members DSIHINIT and DSILUCTD only.

Related Statements: ALIASMEM, NLDM.CDRMDEF, LUC.CTL, CNMTARG, CTL, LOGINIT, R, REPORTS, TASK, W, PARTNER

END

The END statement stops the processing of a sample. No processing takes place after the END statement is encountered. If you use the END statement in a member imbedded using the %INCLUDE statement, processing of both the included member and the member that contains the %INCLUDE statement is stopped.

The syntax for the END statement is:

END

Where:

END Indicate that processing is to stop.

Usage Note: The END statement has no operands and cannot begin in column 1.

F (Filter)

The F (Filter) statement enables you to associate messages to the message indicator classes you defined with the A statement. Based on the tests you specify, a message indicator can be triggered at the authorized receiver. The test will match on any name that begins with the characters specified. For example, a test of &2 = "ABCD" will match on 'ABCD' as well as 'ABCDEF'. You can also use the NETLOG action of the automation table to trigger message indicators. Code the F statement in DSICNM.

To implement the changes to DSICNM, stop and restart the xxxxxVMT task. If the changes result in additional extents being used, stop and restart the NetView program. Code as many F statements for each message as you like. You can code F statements only for message indicator classes in the range of 1–4.

The syntax for the F statement is:

FILTER
**F (Filter)**

```
F class msgnum &e 'string'
```

Where:

- **F** Indicates the statement name. It must be in column 1 of the statement.

- **class**
  Indicates the message indicator class number and must have a value of 1–4. This value must be in column 5 of the statement. This number corresponds to the class number on the A statement.

- **msgnum**
  Indicates the first parsed element of a message (&1). This value must begin in column 7 of the statement.

**Note:** You can code as many F statements for a message number as you require. The NetView program classifies the message based upon the first true F statement for the message number. A single F statement can contain several comparison expressions. Comparison expressions are separated by commas. All of the comparisons on a single F statement must be true for the classification to be made.

- **modname**
  Indicates the name of an optional module that processes the comparison element (&e) and returns a value that the NetView program compares to 'string'. The *modname* must begin in column 16.

  The NetView program is delivered with the module CNMFANGC, which returns a 1-character string representing the type of node. The type of node depends on the value of &e. If you use the F statement and CNMFANGC, you are limited to 1-character comparisons. You can write other modules to meet your needs, or use CNMFANGC, or both. Your module or CNMFANGC is invoked with the following registers:

  - `1` Address of an operand list
  - `13` Address of the caller’s save area
  - `14` Return address of the calling program
  - `15` Entry-point address of *modname*

  The operand list addressed by register 1 contains five addresses, as shown in Table 6.

<table>
<thead>
<tr>
<th>Word</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MVTADDR</td>
<td>The MVT address.</td>
</tr>
<tr>
<td>2</td>
<td>TXTPTR</td>
<td>The address of the comparison element (&amp;e).</td>
</tr>
<tr>
<td>3</td>
<td>TXTLEN</td>
<td>The address of a 1-byte field containing the length of &amp;e.</td>
</tr>
</tbody>
</table>
Table 6. Addresses in the Operand List (continued)

<table>
<thead>
<tr>
<th>Word</th>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>RESPTR</td>
<td>The address of the field containing the result of <code>modname</code> processing. (For CNMFANGC, this is one of the node types described in Table 7.)</td>
</tr>
<tr>
<td>5</td>
<td>RESLEN</td>
<td>The address of a 1-byte field containing the length of the result (pointed to by RESPTR). This word is set to the length of 'string' and cannot be altered. (CNMFANGC uses this word to blank out RESPTR before putting anything in it.)</td>
</tr>
</tbody>
</table>

As an example, CNMFANGC takes `&e` (a node name) and converts it to one of the one-character node types as shown in Table 7.

Table 7. One-Character Node Types

<table>
<thead>
<tr>
<th>Character</th>
<th>Node Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Host</td>
</tr>
<tr>
<td>E</td>
<td>Local PU</td>
</tr>
<tr>
<td>N</td>
<td>NCP/CA major node</td>
</tr>
<tr>
<td>D</td>
<td>Local LU</td>
</tr>
<tr>
<td>L</td>
<td>Line</td>
</tr>
<tr>
<td>B</td>
<td>Application major node</td>
</tr>
<tr>
<td>C</td>
<td>PU/cluster</td>
</tr>
<tr>
<td>A</td>
<td>Application minor node</td>
</tr>
<tr>
<td>T</td>
<td>LU/terminal</td>
</tr>
<tr>
<td>Y</td>
<td>CDRM major node</td>
</tr>
<tr>
<td>S</td>
<td>Switched major node</td>
</tr>
<tr>
<td>Z</td>
<td>CDRM</td>
</tr>
<tr>
<td>R</td>
<td>Switched PU</td>
</tr>
<tr>
<td>W</td>
<td>CDRSC major node</td>
</tr>
<tr>
<td>Q</td>
<td>Switched LU</td>
</tr>
<tr>
<td>X</td>
<td>CDRSC</td>
</tr>
<tr>
<td>F</td>
<td>Local major node</td>
</tr>
<tr>
<td>BLANK</td>
<td>No node found</td>
</tr>
</tbody>
</table>

This result is then compared to 'string' to determine if an alert should be set up.

`&e`

Is the element selected for comparison to 'string'. The expression must be located in columns 16 through 72. If the expression continues onto successive lines, make a comma the last character on a line. The possible values are:

- `&D` Domain identifier.
- `&O` Operator identifier.
- `&T` Time (in the format `hh:mm:ss`).
The HDRMTYPE from DSITIB. Refer to "Control Blocks" in Tivoli NetView for z/OS Customization: Using Assembler for more information on the message header type.

The number of the element as parsed by the NetView program. This number must be in the range of 1–255. &1 is the message number.

= &e is equal to 'string'.

!= &e is not equal to 'string'.

> &e is greater than 'string'.

>= &e is not greater than 'string'.

< &e is less than 'string'.

<= &e is not less than 'string'.

'string'
Is the value or string you want compared with &e.

continuation expression
Is the continuation of an expression from the preceding line. Begin the continuation statement with an F and then continue the expression in columns 16 through 72.

Usage Notes:
• Messages DSI531I, DSI546I, DSI547I, and DSI559I are routed to all NetView operators who are logged on. If an operator is in a terminal access facility (TAF) session when a message is sent, the message is not sent to the operator’s screen until the TAF session ends. At that time, the filter and alert in the status monitor are set off. Depending on the time that elapsed while the operator was in the TAF session, the circumstances under which the message was issued can change, and the message may no longer be relevant.

• When you code the VTAM function MSGMOD=ON, VTAM inserts a 5-character module name into all IST messages between the message number and the message text. If MSGMOD is coded as ON, the F statements you code for IST messages have unpredictable results.

Related Statements: A

function.autotask
The function.autotask statement defines the autotask for the specified function. This statement is coded in CNMSTYLE.

The syntax for the function.autotask statement is:

function.autotask

(—tower_name—)function.autotask.—function_name—autotask_name—

Where:

tower_name
Specifies the tower name.
function_name

Specifies the function name. The function name can be from 1-15 characters.

autotask_name

Specifies the autotask name.

Usage Notes:

- For example, the following statement in CNMSTYLE defines the autotask DSIMCAOP for the MVScmdMgt tower:

  (MVScmdMGT)function.autotask.mvsCmdMgt = DSIMCAOP

  The MVScmdMgt tower must be enabled or this statement is ignored.

- Autotasks may be defined to the NetView program and the SAF product.

- Refer to CNMSTYLE for specific information on the function.autotask statements.

GHB.TCPANAME

The GHB.TCPANAME statement enables you to specify the TCP name for the DUIDGHB task.

The syntax for the GHB.TCPANAME statement is:

```
GHB.TCPANAME
```

Where:

name

Specifies the TCP name for task DUIDGHB.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE GHB command.

HARDCOPY

The HARDCOPY statement designates the devices that you can use for hardcopy printers. Code this statement in CNMSTYLE. For the changes to take effect, stop and restart the NetView program.

Use as many HARDCOPY statements as you need to define all the hardcopy printers.

Although each operator can be assigned to only one hardcopy printer, several operators can share the same printer. However, if too many operators share the same hardcopy printer, messages for that printer may accumulate, and might not be printed for some time after they are received.

The syntax for the HARDCOPY statement is:
Where:

devicename [...]  
  Specifies the name of the printer as it is defined to VTAM. This name can be in the range of 1–8 characters.

Usage Notes:

• Hardcopy devices must be LU type 0 or LU type 1, or use an LU type 0 or LU type 1 logmode entry. Printers attached to SNA controllers as LU type 1 can use the M32875CS logmode. LU type 2 and LU type 3 printers are not supported.
• 3287 printers models 1, 1C, 2, 2C, 11, and 12 require the SCS feature 9660 when operating with controllers in SNA mode. The 3287 model 1C is supported as a model 1. The model 2C is supported as a model 2. The NetView program supports one color (monochrome) only.
• The NORMQMAX value in the member specified by the SCRNFMT parameter of the DEFAULTS command or the NetView-supplied default, 3000, applies to hardcopy printers. Hardcopy printers can get backlogged because they are slow or because they are out of paper.

Related Statements: NORMQMAX

HELD

The HELD statement enables you to change the color of held messages that are not action messages that appear on the command facility panel. The sample member containing the HELD statement is CNMSCNFT. You can code the HELD statement only once.

The syntax for the HELD statement is:

HELD

Where:

colorf  
  Defines foreground color for held messages. The foreground color must be specified before the background color.

attribute  
  Defines alarm, intensity, and highlight attributes for held messages. Specify the attributes only once for the HELD statement.

ON  
  Makes the color following ON apply to the background of the messages being held. This is a required keyword if you specify only a background color.
**colorb**

Defines background color for held messages.

**Usage Notes:**

- You can also create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the HELD statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - BLACK: The color black
  - BLUE: The color blue
  - RED: The color red
  - PINK: The color pink
  - GREEN: The color green
  - TURQ: The color turquoise
  - YELLOW: The color yellow
  - WHITE: The color white
- The highlight operands are:
  - BLINK: The characters blink.
  - REV: The characters change to reverse video.
  - UND: The characters are underlined.
  - NRM: Normal attributes are used.
- The intensity operands are:
  - HIGH: The characters have high intensity.
  - LOW: The characters have low intensity.
- The alarm operand is:
  - BEEP: When a message is shown, an audible alarm sounds.

**Related Statements:** ACTION, CMDLINE, COLUMNHEAD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

---

**HLL**

The HLL statements define the following preinitialized environments:

- **AD/Cycle**® PL/I
- **AD/Cycle** C

Code these statements in CNMSTYLE.

The syntax for the HLLENV statement is:

**HLLENV**
Where:

**IBMADPLI**

Specifies the AD/Cycle PL/I environment.

**IBMADC**

Specifies the AD/Cycle C environment.

**REGENVS=reg_preinit_env**

Specifies the number of preinitialized environments to be defined immediately. This statement can have a value in the range of 0–99. Environments allocated with REGENVS are retained by NetView in a global pool and are available to preinitialization-enabled programs that you define to run in a preinitialized environment. These programs can run in a preinitialized environment on any subtask where the program could ordinarily run. The default value is 2.

**CRITENVS= crit_preinit_env**

Specifies the maximum number of preinitialized environments that can be allocated exclusively for preinitialization-enabled programs with bit 4 set in HLLOPTS. CRITENVS can have a value in the range of 0–99. The default is 0.

**DEFAULT**

Specifies whether running eligible programs in a preinitialized environment should be the default.

**NOTPREINIT**

Specifies that your preinitialization-enabled programs should not run in a preinitialized environment. NOTPREINIT is the initial value for DEFAULT.

**PREINIT**

Specifies that your preinitialization-enabled programs should run in a preinitialized environment.

**PSTACK=preinit_stack**

Specifies the STACK run-time option value used when building the preinitialized environment. The initial value for PSTACK is 4096 bytes.

**PHEAP=preinit_heap**

Specifies the HEAP run-time option value used when building the preinitialized environment. The initial value for PHEAP is 4096 bytes.

**Usage Notes:**

- For more information, refer to the online help for the HLLENV command.
- Refer to [Tivoli NetView for z/OS Customization: Using PL/I and C](#) for more information about HLLOPTS.
HOLDPCNT

The HOLDPCNT statement defines the percentage of the screen allocated for held and action messages. The percentage is based on the maximum area the NetView program allows for held and action messages. The sample member containing the HOLDPCNT statement is CNMSCNFT.

The syntax for the HOLDPCNT statement is:

```
HOLDPCNT

100
```

Where:

- **100**\(\text{nnn}\)
  - Indicates the percentage of the screen allocated for held and action messages. The valid range is 0–100. Zero prevents held and action messages from being retained on the screen. The default is 100.

Usage Notes:

- You can also create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- Message DSI151I is displayed if there is not enough space on the panel to display all of the pending held messages.
- If you specify HOLDPCNT 100, the NetView program reserves 10 lines below the held and action area for normal messages to display. If you specify HOLDPCNT 0, message DSI151I displays to indicate that held or action messages require attention. The messages themselves are not displayed. If you specify HOLDPCNT 1, the NetView program reserves a 2-line area on the screen to allow a message prefix and the first line of a held or action message to display.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

HOLDWARN

The HOLDWARN statement enables you to change the color of message DSI151I, indicating that the held or action area that appears on the command facility screens is full. The sample member containing the HOLDWARN statement is CNMSCNFT. You can code the HOLDWARN statement only once.

The syntax for the HOLDWARN statement is:
Where:

**colorf**
Defines foreground color for message DSI151I. The foreground color must be specified before the background color.

**attribute**
Defines alarm, intensity, and highlight attributes for message DSI151I. You can specify attributes only once for the HOLDWARN statement.

**ON**
Makes the color following ON apply to the background of message DSI151I. This is a required keyword if you specify only a background color.

**colorb**
Defines background color for message DSI151I.

**NONE**
Prevents message DSI151I from appearing on the screen.

**Usage Notes:**

- To create a member containing screen format definitions, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the HOLDWARN statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - **BLACK** The color black
  - **BLUE** The color blue
  - **RED** The color red
  - **PINK** The color pink
  - **GREEN** The color green
  - **TURQ** The color turquoise
  - **YELLOW** The color yellow
  - **WHITE** The color white
- The highlight operands are:
  - **BLINK** The characters blink.
  - **REV** The characters change to reverse video.
  - **UND** The characters are underlined.
  - **NRM** Normal attributes are used.
- The intensity operands are:
  - **HIGH** The characters have high intensity.
  - **LOW** The characters have low intensity.
- The alarm operand is:
  - **BEEP** When a message is shown, an audible alarm sounds.
The idleParms statement specifies operator idle times that are used to determine which operators should be stopped. Code this statement in CNMSTYLE.

The syntax for the idleParms statement is:

```
idleParms
```

Where:

- **idlemin** = minutes
  Specifies the maximum number of minutes that an operator can be idle. Idle operators that are exceeding this time when IDLEOFF runs are logged off, unless previously excepted. The minutes value must be a positive integer.

- **frequency** = minutes
  Specifies how often the IDLEOFF process is invoked.

- **exceptOp** = operatorID
  Specifies a list of operator IDs that are not to be logged off.

- **exceptAuto** = autotask_type
  Specifies which autotasks are exempt from IDLEOFF.
  - **ALL**
    Specifies that all autotasks are exempt.
  - **DIST**
    Specifies distributed autotasks. Those receiving RMTCMD commands are exempt, but system consoles are not exempt as a group.
  - **CONSOLE**
    Specifies that system console autotasks are exempt, but distributed autotasks are not exempt as a group.
  - **NONE**
    Specifies that all attended autotasks are eligible.

- **exceptNNT**
  Specifies which NNT tasks are exempt from IDLEOFF.
  - **ALL**
    Specifies that all NNT tasks are exempt.
idleParms

<table>
<thead>
<tr>
<th></th>
<th>NONE</th>
<th>Specifies that all NNT tasks are eligible. This is the initial setting.</th>
</tr>
</thead>
<tbody>
<tr>
<td>exceptRmtCmd</td>
<td>ALL</td>
<td>Specifies that all users of RMTCMD are exempt, but does not exempt distributed autotasks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This exempts any task that used RMTCMD, even if the task has no current RMTCMD sessions, and even if the RMTCMD failed to start a session.</td>
</tr>
<tr>
<td></td>
<td>NONE</td>
<td>Specifies that all RMTCMD users are eligible. This is the initial setting.</td>
</tr>
<tr>
<td>exceptLU = luname</td>
<td>Specifies a list of LU names that are not to be logged off.</td>
<td></td>
</tr>
</tbody>
</table>

**Usage Note:** You can also use the IDLEOFF command to examine operator idle times. For more information, refer to the [Tivoli NetView for z/OS Command Reference](https://www.ibm.com/support/docview.ws?id=doc15191) or the NetView online help.

IMDAREA

The IMDAREA statement enables you to change the color of the immediate messages that appear on the command facility panel. The sample member containing the IMDAREA statement is CNMSCNFT. Code the IMDAREA statement only once.

The syntax for the IMDAREA statement is:

```
IMDAREA
```

Where:

- **color**
  Defines foreground color for the immediate messages. You must specify the foreground color before the background color.

- **attribute**
  Defines alarm, intensity, and highlight attributes for the immediate messages. You can specify attributes only once for the IMDAREA statement.

- **ON**
  Makes the color following ON apply to the background of the immediate messages. This is a required keyword if you specify only a background color.

- **colorb**
  Defines background color for the immediate messages.

**Usage Notes:**
- To create a member containing screen format definitions, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
IMDAREA

- The NetView-supplied defaults for the IMDAREA statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off

- The color operands are:
  - BLACK The color black
  - BLUE The color blue
  - RED The color red
  - PINK The color pink
  - GREEN The color green
  - TURQ The color turquoise
  - YELLOW The color yellow
  - WHITE The color white

- The highlight operands are:
  - BLINK The characters blink.
  - REV The characters change to reverse video.
  - UND The characters are underlined.
  - NRM Normal attributes are used.

- The intensity operands are:
  - HIGH The characters have high intensity.
  - LOW The characters have low intensity.

- The alarm operand is:
  - BEEP When a message is shown, an audible alarm sounds.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

%INCLUDE

The %INCLUDE statement specifies a member (or a file) to be included by the file in which the %INCLUDE statement appears. The member that is embedded is treated as a part of the member in which it is included. It is embedded at the point at which the %INCLUDE statement appears. Code %INCLUDE statements in DSICMD, DSIOPF, CNMSCNFT, CNMSTYLE, command facility panel-format definition members, command authorization tables, NetView span tables, and the HELPMAP.

You can also use %INCLUDE statements in automation tables, in code point tables, and in members read using NetView disk services with the INCL option. For information about disk services in assembler, see DSIDKS in Tivoli NetView for z/OS Customization: Using Assembler. For information about disk services in PL/I and C, see CNMMEMR in Tivoli NetView for z/OS Customization: Using PL/I and C.

You can use the Data REXX function to conditionally include a member or file using the %INCLUDE statement. For more information about Data REXX, refer to Tivoli NetView for z/OS Customization: Using REXX and the NetView Command List Language.

The syntax for the %INCLUDE statement is:
%INCLUDE

%INCLUDE

---%INCLUDE member---

Where:

%INCLUDE

% must be the first nonblank character on a line. The record cannot be split between lines. INCLUDE must immediately follow the % to ensure that the % is not just a stray percent sign. A space is required after %INCLUDE, and the word INCLUDE must be in all capital letters.

member

Specifies the name of the member to be included. The member name in the %INCLUDE statement begins with the first nonblank character following the word INCLUDE. The member name cannot include blanks and must follow the MVS member naming conventions. If the member name is longer than eight characters, the %INCLUDE statement is rejected.

Usage Notes:

- When using the %INCLUDE statement in members such as DSICMD that expect an END statement at the end of the member, the member that you name on the %INCLUDE statement cannot contain an END statement. If it does, processing stops at the END statement and does not return to the original member.

- If you use the INCL option of DSIDKS to read a file with a %INCLUDE statement in it, the member name on the %INCLUDE statement can be a command list variable. You can use a command list variable for the member name on the %INCLUDE statement when the automation table is loaded by the AUTOTBL command or when the REFRESH command is used to load the command authorization table or NetView span table; a command list variable cannot be used for the member name during initialization but a system symbolic can be used. A command list variable cannot be used for the member name on the %INCLUDE statements in DSIOPF, DSICMD, and code point tables.

  If the member name begins with an ampersand (&), it is considered a command list variable and must use the command list variable naming conventions. A command list variable with a null value is treated as a nonexistent variable.

  When a command list variable used on a %INCLUDE statement is evaluated, the local dictionary is checked first if the disk services read is issued from the REXX environment or NetView command list language.

  A disk services read is issued from the REXX environment or NetView command list language if the command processor that issues the read is called from a REXX or NetView command list language command processor.

  If the disk services read is not issued from the REXX environment or NetView command list language a local dictionary does not exist. If the variable cannot be found in the local dictionary, or if it has a null value, disk services checks the task dictionary. If the variable is not found in the task dictionary, or if it has a null value, disk services checks the common global dictionary. If the variable is not found in the common global dictionary, or if it has a null value, disk services checks the system symbolics. If the variable is not found in the system symbolics or has a null value the member name is treated as not valid. Empty members are treated as members that are not valid.

- The member name in MVS is the name of a member in the DD in which the member containing the %INCLUDE statement resides. Extra characters after the
member name on the %INCLUDE statement are ignored. If the member specified cannot be found, or if the %INCLUDE statement contains a syntax error, the statement is rejected.

- If you are using the %INCLUDE statement with NetView disk services and the %INCLUDE statement is rejected, the disk services return code indicates what the error is and returns the %INCLUDE statement that is not valid instead of the next logical record. If you use a command list variable for the member name and the %INCLUDE statement is not valid, the %INCLUDE statement is returned with the variable substitution made in the place of the next logical record to be read. If the %INCLUDE statement with the variable substitution exceeds the record length, the statement is truncated.

**INDENT**

The INDENT statement defines the indentation characteristics for single text messages. The sample member containing the INDENT statement is CNMSCNFT.

The syntax for the INDENT statement is:

```plaintext
INDENT
```

Where:

**11|nn**

 Defines the number of characters of indentation allowed for continued text lines. This is a decimal number. The maximum value allowed is 75. The NetView-supplied default is 11.

**NEWLINE**

 Specifies that the text begins on the line after the prefix for single line messages. The text is indented using the value of `nn`. This prevents having a small amount of text at the end of a long prefix. Also, longer messages are displayed as paragraphs.

**Usage Note**: To create a member containing screen format definitions, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

**Related Statements**: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

**INIT**

The INIT statement starts a NetView function during initialization of the NetView program. Code this statement in CNMSTYLE.

The syntax for the INIT statement is:
**INIT**

**INIT** function = \[ \begin{array}{c}
\text{NO} \\
\text{YES}
\end{array} \]

Where:

- **function**
  - Specifies the function to be started:
    - NRM (NetView Resource Manager)
    - TIMER
  - **NO**
    - Indicates not to start the specified function.
  - **YES**
    - Indicates to start the specified function.

**Usage Notes:**
- You can start the NetView Resource Manager at a later time by issuing the INITNRM command.
- You can restore the timer command data at a later time by issuing the RESTORE TIMER command. For more information, refer to the *Tivoli NetView for z/OS Command Reference* or the NetView online help.

**inStore**

The inStore statement adds in-storage members. The members are then read from storage rather than the using disk services. Code this statement in CNMSTYLE.

The syntax for the inStore statement is:

```
inStore
```

Where:

- **dataset_name**
  - Specifies the data set name.
- **member_name**
  - Specifies the member name.
  - You can specify *NONE* to prevent any members from being cached for the specified DD name.

**Usage Notes:**
- The inStore statement takes effect before any tasks are started.
- You can also use PIPE INSTORE to add in-storage members. For more information, refer to the *Tivoli NetView for z/OS Customization: Using Pipes* or the NetView online help.
The IPLOG statement defines TCP/IP definitions for the syslog server (task DSIIPLOG, member DSIILGCF). Code this statement in CNMSTYLE.

The syntax for the IPLOG statement is:

```
IPLOG
```

Where:

- **TCPANAME** = name
  
  Defines the TCP/IP address space name.

- **PORT** = port
  
  Defines the port number on which DSIIPLOG waits for connection requests.
  
  The default is 514.

- **SOCKETS** = number
  
  Specifies the maximum number of simultaneous users. The default is 100.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE IPLOG command.

---

The ISPAN statement sets up the initial span of control for the operator when SECOPTS.OPSPAN = NETV is specified in CNMSTYLE or OPSPAN=NETV is specified on the REFRESH command. The operator can remove this span of control by entering the NetView STOP command with the SPAN operand. Code this statement in the DSIPRF member specified by the PROFILEN statement associated with the operator.

You can use ISPAN as often as necessary to list all the span names. Changes made to the ISPAN statement take effect the next time an operator logs on to the NetView program using the profile containing the statement. See “OPERATOR” on page 151 and “PROFILEN” on page 161 to determine how a PROFILEN statement is associated with an operator.

The syntax for the ISPAN statement is:

```
ISPAN
```
ISPAN

Where:

label
Indicates the optional label for the ISPAN statement. This label identifies the statement in any related error messages.

spanname [...]
Indicates the 1–8 character name of the span. Code as many spannames and as many ISPAN statements as necessary to specify all spans-of-control.

R Specifies that an access level of READ is to be granted to spanname. This allows information-only access to resources and views defined to spanname. This level includes functions such as LIST and DISPLAY.

U Specifies that an access level of UPDATE is to be granted to spanname. This allows change access to resources and views defined to spanname. This level includes functions such as VARY and the generic activate action on the NetView management console pull-down.

C Specifies that an access level of CONTROL is to be granted to spanname. This allows multi-read and single-write access to resources and views defined to spanname.

A Specifies that an access level of ALTER is to be granted to spanname. This allows multi-write access to resources and views defined to spanname. This is the default if no access level is specified.

Related Statements: AUTH, OPERATOR, SECOPTS, PROFILE, PROFILEN, SPAN

JesJobLog

The JesJobLog statement enables you to allocate the JES job log. The JesJobLog statement is coded in CNMSTYLE.

The syntax for the JesJobLog statement is:

```
JesJobLog = YES
```

Where:

YES Specifies to allocate the JES job log. This is the default.

NO Specifies that the JES job log will not be allocated.

KCLASS

The KCLASS statement defines keep classes. A keep class determines what data is kept for the sessions mapped. A session is mapped into a keep class when the session starts. Once a session is mapped into a keep class, it remains there until the session ends.

Code one KCLASS statement for each keep class you define. You can code as many KCLASS statements as necessary.
If you code KCLASS and MAPSESS statements, create a member in which to put them. The name for this new member must match the name coded on the NLDM.KEEPMEM statement in CNMSTYLE. The KCLASS statements are the first statements in the member.

The KCLASS and MAPSESS statements control the amount of session awareness data recorded. If you do not code any of these statements, data for all sessions is recorded according to the defaults for the KCLASS statement.

The syntax for the KCLASS statement is:

**KCLASS**

\[ \textit{cname} \text{ KCLASS} \]

- `AVAIL`: YES or NO
- `DASD`: YES or NO
- `DASD` options include:
  - `BINDFAIL`, `DATA`, `FAILURES`, `INITFAIL`, `RTMDATA`
  - `SESSFAIL`, `SESSNORM`, `TRACDATA`
- `DGROUP`: (name)
- `RENAMING`: PRI, SEC
- `KEEPIU`: 7
- `KEEPPIU`: nnn
- `KEEPSESS`: 0 or nnn
- `SAW`: YES or NO
- `SAW` options include:
  - `1`, `DATA`, `FAILURES`, `INITFAIL`, `RTMDATA`
  - `SESSFAIL`, `SESSNORM`, `TRACDATA`

**Notes:**

1. There are limitations on specifying YES with the DASD keyword; see the descriptive text.

**Where:**

- `cname` indicates the name of the defined keep class. This name is used on one or more MAPSESS statements as the value of `cname` in the KCLASS operand. The name must be in the range of 1-8 characters and must start in column 1 of the statement. This is a required operand.
KCLASS

**AVAIL=**YES | NO

Defines whether availability data is kept for the sessions mapped into this keep class.

**YES**  Specifies that availability data is kept

**NO**  Specifies that availability data is not kept

The default corresponds to the value of the NLDMS.ESSTATS statement in CNMSTYLE. If NLDM.ESSTATS is AVAIL, the default for AVAIL is YES. If NLDM.ESSTATS is YES or NO, the value for AVAIL is ignored. Table 8 shows combinations of values and their meanings.

**Table 8. Combinations of AVAIL and SESSTATS**

<table>
<thead>
<tr>
<th>If SESSTATS=</th>
<th>Then KCLASS</th>
<th>AVAIL=</th>
<th>This means that:</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAIL</td>
<td>Defaults to YES</td>
<td>Availability data is logged.</td>
<td></td>
</tr>
<tr>
<td>YES</td>
<td>Is ignored</td>
<td>Both availability data and accounting data are logged.</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>Is ignored</td>
<td>Neither availability data nor accounting data is logged.</td>
<td></td>
</tr>
</tbody>
</table>

**DASD**

Defines the conditions for recording the sessions mapped into this keep class to the VSAM file.

**NO**  Does not record sessions to the VSAM file. This is the default.

**YES**  Records all of the following specified sessions to the VSAM file. If you choose some of the sessions (not all), leave YES out of the statement. When you code some sessions within the parentheses, YES is assumed for those sessions and you do not need to code it. If you code YES with some of the sessions, message AAU234I is issued.

**BINDFAIL**

Records sessions to the VSAM file if there is BIND failure.

**DATA**

Records sessions to the VSAM file only if there is some trace or RTM data to record with the session.

**FAILURES**

Records LU-LU sessions to the VSAM file for the following occurrences:

- An abnormal UNBIND reason code
- BIND failure
- A failure to set up prior to BIND flow (INIT failure)

**INITFAIL**

Records sessions to the VSAM file if there is a failure to set up prior to BIND flow (INIT failure).

**RTMDATA**

Records sessions to the VSAM file if there is some RTM data associated with the session.

**SESSFFAIL**

Records sessions to the VSAM file if there is an abnormal UNBIND reason code.
SESSNORM
Records sessions to the VSAM file if there is a normal UNBIND reason code.

TRACDATA
Records sessions to the VSAM file if there is some trace data associated with the session.

DGROUP
Specifies the grouping characteristics of all the MAPSESS sessions mapping to this KCLASS statement. You can group sessions under a user-supplied name, or you can defer the DGROUP name until the session ends by using the *PRI or *SEC values. Deferring the DGROUP name enables the definition of multiple DGROUPs with a single KCLASS statement, with the DGROUP name being either the primary or secondary session partner name.

name
Indicates the user-supplied name of the group. name must consist of valid resource name characters (A–Z, 0–9, @, #, $). You cannot specify wildcard characters (* and ?) for name.

RENAME
Renames either the primary or secondary session partner name to the group name before recording the session. RENAME has no effect when the current global KEEPSESS value or session KEEPSESS value is zero (0).

PRI
Renames the primary session partner name to the group name. PRI is the default.

SEC
Renames the secondary session partner name to the group name. PRI and SEC are mutually exclusive. SEC has no effect when you do not specify RENAME.

*PRI
Indicates that the DGROUP name is to be set to the primary session partner name.

*SEC
Indicates that the DGROUP name is to be set to the secondary session partner name.

KEEPPIU=7|nnn
Determines the initial PIU keep count for the sessions mapped into this keep class. The value of nnn must be in the range of 1–999. The default is 7. Once a session is active, you can change the number of PIUs kept for that session dynamically using the KEEP PIU command.

KEEPSESS=0|nnn
Indicates the DASD session wrap count (0–999) for all sessions mapping into this KCLASS. If the value is zero (0), session wrapping does not occur until the count of sessions for this KCLASS exceeds 32767. Use the keyword DASD=NO to prevent recording of sessions for this KCLASS. If KEEPSESS is not coded, the global KEEPSESS value is used for sessions mapping into this KCLASS. If the global wrap count in DSIPARM member AAUPRMLP is zero (0), wrapping does not occur, regardless of the value of KEEPSESS. Also, sessions will not be recorded by DGROUPs. 0 is the default.
**KCLASS**

**SAW=**YES | NO

Defines whether session awareness (SAW) data should be kept for the sessions mapped into this keep class.

- **YES** Keeps session awareness data. This is the default.
- **NO** Does not keep session awareness data.

Session awareness data is always kept for SSCP-SSCP sessions and SSCP-PU sessions. To ensure that session awareness data is kept for all sessions, including SSCP-LU and LU-LU sessions, specify SAW=YES. To collect response time monitor (RTM) data for LUs attached to terminal control units equipped with the RTM feature, you need to keep session awareness data for both SSCP-LU and LU-LU sessions.

Keep session awareness data for all cross-network sessions. This ensures that cross-network data retrieval functions work correctly.

**Usage Notes:**

- Use commas and parentheses when you select more than one of the DASD options.
- Sessions are always recorded when an operator issues a FORCE command, regardless of how you specify DASD. For additional information about the FORCE command, refer to NetView online help.

Keep session awareness data for all cross-network sessions. This ensures that cross-network data retrieval functions work correctly.

- If you code NLDM.SAW=NO in CNMSTYLE, the NetView program does not receive session awareness data from VTAM until the ENABLE command is issued.

**Related Statements:** NLDM.SAW, MAPSESS, PCLASS

**KEYCLASS**

The KEYCLASS statement is obsolete. It is used for migration purposes only. For more information, refer to the Tivoli NetView for z/OS Installation: Migration Guide.

**Related Statements:** CMDCLASS, OPCLASS, VALCLASS

**LASTLINE**

The LASTLINE statement enables you to change the color of the dashed line that separates new and old messages on the command facility panel. The sample member containing the LASTLINE statement is CNMSCNFT. Code the LASTLINE statement only once.

The syntax for the LASTLINE statement is:

**LASTLINE**

```
LASTLINE  \colorf \attribute \ON \colorb
```

**Where:**
colorf
Defines foreground color for the last line. The foreground color must be specified before the background color.

attribute
Defines alarm, intensity, and highlight attributes for the last line. You can specify attributes only once for the LASTLINE statement.

ON
Makes the color following ON apply to the background of the last line. This is a required keyword if you specify only a background color.

colorb
Defines background color for the last line.

Usage Notes:
• To create a member containing screen format definitions, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
• The NetView-supplied defaults for the LASTLINE statement are:
  - Turquoise foreground
  - Black background
  - Normal highlighting
  - Low intensity
  - Alarm off
• The color operands are:
  BLACK  The color black
  BLUE   The color blue
  RED    The color red
  PINK   The color pink
  GREEN  The color green
  TURQ   The color turquoise
  YELLOW The color yellow
  WHITE  The color white
• The highlight operands are:
  BLINK  The characters blink.
  REV    The characters change to reverse video.
  UND    The characters are underlined.
  NRM    Normal attributes are used.
• The intensity operands are:
  HIGH   The characters have high intensity.
  LOW    The characters have low intensity.
• The alarm operand is:
  BEEP   An audible alarm sounds.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

LOADEXIT
The LOADEXIT statement enables you to specify which installation exits you want to load. LOADEXIT statements are coded in CNMSTYLE. The NetView program attempts to load only the exits specified. Code one LOADEXIT statement for each installation exit to be loaded.
**LOADEXIT**

The syntax for the LOADEXIT statement is:

```
LOADEXIT

exit_name = YES | NO

Where:

- `exit_name` indicates one of the following installation exits:
  - DSIEX01
  - DSIEX02A
  - DSIEX03
  - DSIEX04
  - DSIEX05
  - DSIEX06
  - DSIEX07
  - DSIEX09
  - DSIEX10
  - DSIEX11
  - DSIEX12
  - DSIEX13
  - DSIEX14
  - DSIEX16
  - DSIEX16B
  - DSIEX17
  - DSIEX18
  - DSIEX19
  - DSIEX20
  - DSIEX21

- **YES** specifies to load the exit.
- **NO** specifies not to load the exit. This is the default.

**Usage Notes:**
- DSIEX15 is an obsolete exit, but is supported if used in your environment.
- Refer to the *Tivoli NetView for z/OS Customization: Using Assembler* for information on installation exits.

**LOCKIND**

The LOCKIND statement enables you to change the color of the lock indicator (***) that is displayed on the command facility panel. The sample member containing the LOCKIND statement is CNMSCNFT. Code the LOCKIND statement only once.

The syntax for the LOCKIND statement is:

```
LOCKIND
```
LOCKIND

- \texttt{colorf} - Defines foreground color for the lock indicator. The foreground color must be specified before the background color.
- \texttt{attribute} - Defines alarm, intensity, and highlight attributes for the lock indicator. Specify attributes only once for the LOCKIND statement.
- \texttt{ON} - Makes the color following \texttt{ON} apply to the background of the lock indicator. This is a required keyword if you specify only a background color.
- \texttt{colorb} - Defines background color for the lock indicator.

**Usage Notes:**

- To create a member containing screen format definitions, specify the name of the member on the SCRFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the LOCKIND statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - \texttt{BLACK} - The color black
  - \texttt{BLUE} - The color blue
  - \texttt{RED} - The color red
  - \texttt{PINK} - The color pink
  - \texttt{GREEN} - The color green
  - \texttt{TURQ} - The color turquoise
  - \texttt{YELLOW} - The color yellow
  - \texttt{WHITE} - The color white
- The highlight operands are:
  - \texttt{BLINK} - The characters blink.
  - \texttt{REV} - The characters change to reverse video.
  - \texttt{UND} - The characters are underlined.
  - \texttt{NRM} - Normal attributes are used.
- The intensity operands are:
  - \texttt{HIGH} - The characters have high intensity.
  - \texttt{LOW} - The characters have low intensity.
- The alarm operand is:
  - \texttt{BEEP} - An audible alarm sounds.

**Related Statements:** \texttt{ACTION}, \texttt{CMDLINE}, \texttt{COLUMNHEAD}, \texttt{HELD}, \texttt{HOLDPCNT}, \texttt{HOLDWARN}, \texttt{IMDAREA}, \texttt{INDENT}, \texttt{LASTLINE}, \texttt{MLINDENT}, \texttt{NOPREFIX}, \texttt{NORMAL}, \texttt{NORMQMAX}, \texttt{PREFIX}, \texttt{TITLE}, \texttt{TITLEDATE}, \texttt{TITLEDOMID}, \texttt{TITLEOPID}, \texttt{TITLESTAT}, \texttt{TITLETIME}
The LOGINIT statement defines the initialization operands for the network log, trace logs, and optional NetView sequential logs. The AUTOFLIP operand specifies whether the NetView program should automatically continue recording on the primary data set when the secondary data set becomes full. The RESUME operand specifies whether to begin recording at the beginning of the primary data set or to start after the last entry recorded. For the most efficient logging, set both AUTOFLIP and RESUME to YES.

The syntax for the LOGINIT statement is:

```
LOGINIT
```

```
\texttt{AUTOFLIP=NO|YES}
```

```
\texttt{RESUME=NO|YES}
```

Where:

\textit{label}

Is the optional label for the LOGINIT statement. This label identifies the statement in any related error messages.

\textbf{AUTOFLIP=NO|YES}

Defines whether logging should continue on the primary data set when the secondary data set is full. The value of this operand must be one of the following:

\textbf{NO}

Does not continue writing on the primary data set when the secondary data set is full. NO is the default.

\textbf{YES}

Continues writing at the beginning of the primary data set when the secondary data set is full.

\textbf{RESUME=NO|YES}

Defines whether logging should begin where it left off in either data set, or start at the beginning of the primary data set. The value of this operand must be one of the following:

\textbf{NO}

Begins writing at the beginning of the primary log. NO is the default.

\textbf{YES}

Resumes writing following the last entry recorded in the log data sets.

If you define RESUME=YES and AUTOFLIP=NO, you might need to restart the network log or trace task because of full data sets, depending on where logging left off.

\textbf{Usage Notes:}

- If you allocate only one data set, logging stops when the data set is full. However, if you define a secondary data set, it is used automatically.
- Code this statement in the members specified by the MEM keyword of the DSILog and DSITRACE TASK statements or by task statements that are using NetView sequential log services. It must follow the DSTINIT statements in the member. The sample members supplied with the NetView program are DSILogbk and DSITrcbk.
Changes to the member containing the LOGINIT statement do not take effect until you stop and restart the sequential logging tasks, including the DSILOG and DSITRACE tasks. If changing the member results in additional extents being used, you need to stop and restart the NetView program before the changes take effect.

Related Statements: TASK

LOGONPW

The LOGONPW statement enables you to specify whether the NetView logon screen can be bypassed by operators specifying their password as part of the VTAM logon specification. This statement is coded in CNMSTYLE.

The syntax for the LOGONPW statement is:

```
LOGONPW= NO
YES
```

Where:

NO Specifications that operators cannot bypass the NetView logon screen.
YES Specifications that operators can bypass the NetView logon screen.

LU

The LU statement translates LU names in interconnected networks. It assigns alias names by which logical unit names in other networks are known in this network. Code this statement in the members specified on the ALIASMEM statement. The sample member supplied with the NetView program is DSIALTAB.

The syntax for the LU statement is:

```
LU label targname,targnet,origname,sscpname
```

Where:

label
Indicates the optional label for the LU statement. This label identifies the statement in any related error messages.

targname
Indicates the real name of the logical unit as it is defined to the VTAM in this domain.

targnet
Indicates the 1–8 character name of the network where the logical unit is known by its real name.

origname
Indicates the alias translation name for this logical unit.
sscpname

Indicates the 1–8 character name of the SSCP that owns the real logical unit in the target network. sscpname is optional. If you do not code sscpname, the SSCP identifier is taken from the VTAM CDRSC definition statement.

Related Statements: ALIASMEM, ORIGNET

LUC

The LUC statements define LUC initialization parameters for the CNM data transfer task domidLUC. These parameters are used by member DSILUCTD. Code these statements in CNMSTYLE.

The syntax for the CNM statement is:

LUC

Where:

LUC.CNMTARG.suffix = luname [YES | NO]

The LUC.CNMTARG statement defines authorized LU names for the CNM data transfer task.

suffix The suffix consists of 1–8 characters and must be unique (A,B,C in the example).

LUC.CNMTARG.A
LUC.CNMTARG.B
LUC.CNMTARG.C

luname Indicates the 4-8 character application (APPL) name for the remote CNM data transfer task. CNM data transfer tasks are named xxxxLUC, where xxxx is the NetView program identifier of the NetView program where the CNM data transfer task runs. You can have multiple LUC.CNMTARG statements. To view session data from another NetView program, you must be authorized by that NetView program.

YES Indicates that the session is persistent and remains active regardless of the time elapsed between conversations between this NetView program and a remote NetView program. A persistent session is one that remains active regardless of the time elapsed between conversations involving this NetView program and a remote NetView program. A nonpersistent session is one that is ended if it is inactive for longer than the value specified in the nonpersistent sessions timeout interval in DSICTMOD.

NO Indicates that the session is nonpersistent and ends if the time elapsed between conversations between this NetView program and a remote
NetView program reaches the value of the nonpersistent sessions time-out constant. Refer to "Centralizing Operations" in the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information.

**LUC.CTL = [GLOBAL | SPECIFIC]**

The LUC.CTL statement defines the authorization requirements for the CNM data transfer sessions.

**GLOBAL**

Indicates there is no authority check when the NetView program establishes new NNT cross-domain session monitor sessions. The system accepts any session initiation request. This is the initial setting in CNMSTYLE.

**SPECIFIC**

Indicates that sessions are initiated and accepted only for specific LUs. Specific LUs are those that are defined on the LUC.CNMTARG statement.

**LUC.MAXSESS = number**

The LUC.MAXSESS statement specifies the number of parallel sessions the LUC task can establish.

*number*

Indicates the number of parallel sessions that can be established. The maximum number is 65535. The default is 20.

**LUC.PERSIST = [YES | NO]**

The LUC.PERSIST statement specifies the default PERSIST setting.

Use the LUC.PERSIST statement to specify whether a session between this NetView program and a remote NetView program is persistent. The default value of the timeout interval is zero. Refer to "Centralizing Operations" in the Tivoli NetView for z/OS Installation: Configuring Additional Components for information about changing the value of the timeout interval.

**YES**

Specifies that the LUC sessions are persistent. This is the default.

**NO**

Specifies that the LUC sessions are nonpersistent and will unbind in the number of seconds specified in DSICTMOD.

**Usage Notes:**

- If you specify GLOBAL on the LUC.CTL statement, the NetView program ignores the specific LU names specified in the LUC.CNMTARG statements. If you specify SPECIFIC on the LUC.CTL statement, a CNMTARG statement is required for each remote NetView program that communicates with this NetView program. You define an LU name only to start a session with that LU.
- To implement member changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE LUC command.

**Related Statements:** CDRMDEF, DOMAINS, LU, TASK.

---

**M (Maximum)**

The M statement allows you to set maximum values for controlling the frequency and number of times that NetView issues certain VTAM commands that are used by the status monitor. If you do not code this statement, these commands will be issued by the NetView program at default time intervals with no limit on the number of times the commands are issued. You code this statement in DSIPARM.
member DSICNM. Changes to DSICNM do not take effect until you stop and restart the xxxxVMT task. If changing the member results in additional extents being used, you need to stop and restart the NetView program before the changes take effect.

The syntax for the M statement is:

**M (Maximum)**

```
M
  MAXREACT
  00
  react_count
  00
  react_interval
  comments
```

Where:

**M** is the statement name. It must be in column 1. MAXREACT|REACTINT is the name of the maximum value to be set. This must begin in column 3 of the statement.

**MAXREACT**

Specifies the maximum number of times that the status monitor will attempt reactivation of a resource. The default value 00 results in no limit.

**REACTINT**

Specifies the number of minutes that you want used as the interval for status monitor reactivation attempts. The default value 00 results in the status monitor reactivation attempts being done every minute.

**react_count**

Specifies the number of times that status monitor reactivation is to be attempted. The default of 00 results in no limit. If specified, this value must begin in column 18.

**react_interval**

Specifies the number of minutes between status monitor reactivation attempts. If a value of 00 is specified explicitly or by default, reactivation is attempted once every minute. If specified, this value must begin in column 18.

**comments**

Specifies any comments which you would like to add as documentation.

**Usage Note:** The value specified on the M statement applies to all resources being monitored by the status monitor.

### MAPSESS

You can define a series of MAPSESS statements to describe sessions that are mapped into each performance or keep class. A session is mapped into a performance or keep class when the configuration of that session matches all the operands of a MAPSESS statement. A session can match more than one MAPSESS statement. If it does, it is mapped by the first matching MAPSESS statement.
For interconnected networks, alias names are used for resources that are not in your local network. If you want to map resources not in your local network, you need to specify the alias names instead of the real names on your MAPSESS statements.

For sessions where explicit route (ER), virtual route (VR), or transmission priority (TP) data are not available to the NetView program, the session is assumed to match these operands.

This sequence of statements is in the member named on the NLDM.PERFMEM or NLDM.KEEPMEM statement in CNMSTYLE.

The syntax for the MAPSESS statement is:

```
MAPSESS

label

MAPSESS PCLASS= cname

KCLASS= cname

PRI= pname

SEC= sname

ER= nn

VR= n

TP= n
```

**Where:**

- **label**
  Indicates the optional label that identifies the MAPSESS statement in any related error messages.

- **PCLASS= cname**
  Names the performance class, as specified on the PCLASS statement, to which sessions that match all the other MAPSESS operands are assigned. This operand is required.

- **KCLASS= cname**
  Names the keep class, as specified on the KCLASS statement, to which sessions that match all the other MAPSESS operands are assigned. This operand is required.

- **PRI= pname**
  Indicates the name of the primary end point of any session that matches this MAPSESS statement. If the desired primary end point is the SSCP, use the name specified on the SSCPNAME start option for VTAM. If you have not specified an SSCPNAME, use VTAM for the name of the SSCP. If you omit the PRI operand, or code PRI= *, all primary endpoint names match this MAPSESS statement.

- **SEC= sname**
  Names the secondary end point of any session that matches this MAPSESS statement. If the desired secondary end point is a cross-domain SSCP, specify the CDRM name for this SSCP. If you omit the operand, or code SEC= *, all secondary end point names match this operand.

  To vary one character, use a question mark (?). For example, A?B matches any name that begins with A, ends with B, and has one character between, such as AAB, ABB, AXB, and so on. A??B matches any name that begins with A, ends with B, and has any two characters between, such as AAXB. A character must
always appear in the position of a ?; that is, A?B would not be matched by AB because there is no character replacing the character ?.

To vary a string of characters at the end of a group of resources, use an asterisk (*). For example, TSO* matches any name that begins with the letters TSO, such as TSOXYZ, TSOB2219, and so on. You can use the * only at the end of a character string. You cannot use it between characters.

**ER=n**
Indicates the explicit route number required for a session to match this MAPSESS statement. If you code a value, it must be in the range of 0–15. If you omit the operand, or code ER=*, all ER values match this operand.

**VR=n**
Indicates the virtual route number required for a session to match this MAPSESS statement. If you code a value, it must be in the range of 0–7. If you omit the operand, or code VR=*, all VR values match this operand.

**TP=n**
Indicates the transmission priority required for a session to match this MAPSESS statement. If you code a value, it must be in the range of 0–2. If you omit the operand, or code TP=*, all TP values match this operand.

**Usage Note:** You can code the PRI and SEC names with special characters so that a group of primary or secondary endpoints match the same MAPSESS statement.

**Related Statements:** KCLASS, PCLASS

---

**MCON**

The MCON statement defines TCP/IP definitions for the 3270 management console (task DSITCPIP, member DSITCPCF). Code this statement in CNMSTYLE.

The syntax for the MCON statement is:

```
MCON
```

**Where:**

**TCPANAME = name**
Defines the TCP/IP address space name.

**PORT = port**
Defines the port number on which to wait for connection requests. The default is 9999. This is also defined to the client.

**SOCKETS = number**
Specifies the maximum number of simultaneous users in the range 50–1000. The default is 50.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE MCON command.
memStore

The memStore statement manages an algorithm that loads the NetView PDS members with the highest usage in storage to avoid additional disk I/O and CPU usage when accessing PDS members from the NetView program. Code the memStore statement in CNMSTYLE.

The syntax for the memStore statement is:

```
memStore

  stgLimit = storg
  minHits = usage_hits
  frequency = minutes
  never = ddname, membername
```

Where:

- `stgLimit = storg`
  Specifies the percentage of your region size above 16 M. This is the amount of storage allocated to in-storage members managed by memStore. The `storg` value must be positive. The default is 5%.

- `minHits = usage_hits`
  Specifies the minimum number of usage hits against the member. A member with less than this will not be loaded into storage.

- `frequency = minutes`
  Specifies the time interval in minutes that memStore should test for usage.

- `never = ddname, membername`
  Specifies `ddname` and any associated `membernames` that memStore should not cache. If `ddname` is not specified (or if an asterisk is used), the action is against that member name in all DD data sets defined to the NetView program. Use a period to separate `ddname` from `member name`.

Usage Notes:

- To disable the memStore function, specify `memStore.stgLimit = 0%`.
- Use the MEMSTOUT command to control or refresh members.
- For more information, refer to the memStore statement in CNMSTYLE and to the Tivoli NetView for z/OS Command Reference.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE MEMSTORE command.

MLINDENT

The MLINDENT statement defines the indentation characteristics for each line of a multiline write-to-operator (MLWTO) message after the MLWTO prefix line. The sample member containing the MLINDENT statement is CNMSCNFT.

The syntax for the MLINDENT statement is:
Where:

\(0 \leq nn\)

Defines the number of characters of indentation allowed for each line of an MLWTO message. This is a decimal number. The maximum value allowed is 75. The NetView-supplied default is zero (0).

Usage Note: You can create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETEME

---

**MODE**

The MODE statement defines equivalent names for duplicate logon mode names in interconnected networks. You code this statement in the members specified on the ALIASMEM statement. The sample member supplied with the NetView program is DSIALTAB.

The syntax for the MODE statement is:

```
MODE label \[targname,targnet,origname\]
```

Where:

**label**

Indicates the optional label for the MODE statement. This label identifies the statement in any related error messages.

**targname**

Indicates the logon mode name that applies to the target network. This name is also called the equivalent name because the MODE name becomes equivalent to the MODE name in the origin network.

**targnet**

Indicates the 1–8 character name of the network where the logon mode is known by its equivalent name.

**origname**

Indicates the name by which the logon mode name is known in the origin network.
The MVSPARM.DEFAUTH statement specifies the default authority for the EMCS consoles that are obtained by the NetView program. You code the MVSPARM.DEFAUTH statement in CNMSTYLE.

The syntax for the MVSPARM.DEFAUTH statement is:

```
MVSPARM.DEFAUTH
```

Where:

**MASTER**
The EMCS console can enter all possible MVS commands. This is the default in CNMSTYLE.

**ALL**
The EMCS console can enter the following command groups:
- Informational
- System control
- I/O control
- Console control

**CONS**
The EMCS console can enter informational and console control command groups.

**CONS&I/O**
The EMCS console can enter the following command groups:
- Informational
- I/O control
- Console control

Enter CONS&I/O without spaces.

**CONS&SYS**
The EMCS console can enter the following command groups:
- Informational
- System control
- Console control

Enter CONS&SYS without spaces.

**INFO**
The EMCS console can enter informational command group commands. It is recommended that you change the DEFAUTH value to INFO, and selectively
permit operators to have higher authorization values by command authorization checking the GETCONID command.

I/O
The EMCS console can enter informational and I/O control command groups.
You must enter I/O without spaces.

I/O&SYS
The EMCS console can enter the following command groups:
- Informational
- System control
- I/O control
Enter I/O&SYS without spaces.

SYS
The EMCS console can enter informational and system control command groups.

Usage Notes:
- The AUTH parameter of the GETCONID command overrides the MVSPARM.DEFAUTH value.
- The AUTH parameter specified using the OPERPARM segment in the resource access control facility (RACF) or an equivalent security access facility (SAF) product overrides the MVSPARM.DEFAUTH value and the GETCONID AUTH value.
- RACF (or an equivalent SAF product) protection of individual commands overrides the authority level of the EMCS console.

MVSPARM.MIGRATE
The MVSPARM.MIGRATE statement specifies whether the EMCS consoles obtained by the NetView program request migration IDs. You code the MVSPARM.MIGRATE statement in CNMSTYLE.

The syntax for the MVSPARM.MIGRATE statement is:

```
MVSPARM.MIGRATE
```

Where:

NO
- Specifies that extended consoles do not request migration IDs. NO is the default.

YES
- Specifies that EMCS consoles request migration IDs.

Usage Notes:
- Use the MVSPARM.MIGRATE=YES statement for operators who do not use the GETCONID command to obtain a console, or who do not use the GETCONID command with the MIGRATE keyword.
You can override the MVSPARM.MIGRATE setting with the GETCONID command or with security software.

**MVSPARM.MSGIFAC**

The MVSPARM.MSGIFAC statement specifies whether the MVS message interface used is the subsystem interface or the extended multiple console support (EMCS). You code the MVSPARM.MSGIFAC statement in CNMSTYLE.

The syntax for the MVSPARM.MSGIFAC statement is:

```
MVSPARM.MSGIFAC
```

Where:

**SYSTEM**

The NetView program uses EMCS consoles. With EMCS consoles, MVS messages are delivered to the EMCS consoles in use by the NetView program instead of over the subsystem interface. The subsystem interface is still used for commands flowing from MVS to NetView autotasks. The task with the load module named CNMCSSIR obtains an EMCS console, which receives all the messages marked AUTO(YES) or AUTO(TOKEN) in the MVS message processing facility (MPF) table.

If you specify SYSTEM, you must also specify `MSGIFAC='SYSTEM'` on the EXEC parameter for the NetView subsystem interface (CNMPSSI) start procedure. SYSTEM is the default.

**CMDONLY**

Specifies to use only the task with the load module named CNMCSSIR for receiving commands from MVS consoles. The task with the load module named CNMCSSIR does not receive and route messages. Use this option if you are using EMCS consoles with MVS route codes for routing messages directly to operators and autotasks. If you specify CMDONLY, you must also specify `MSGIFAC='SYSTEM'` in CNMPSSI.

**QUESSI**

Provides the same function as the USESSI value. Additionally, with this option, messages are queued in the NetView subsystem address space as soon as the subsystem address space completes initialization. With this option, the subsystem interface router task (the task with the load module named CNMCSSIR) does not need to be active in order for messages to be queued in the NetView subsystem address space. Once the router task completes initialization, processing of the queued messages begins. If you specify QUESSI, you must also specify `MSGIFAC='QUESSI'` in CNMPSSI.

**QSSIAT**

Provides the same function as the QUESSI value. Additionally, with this option, messages are queued in the NetView subsystem address space while NetView is inactive. The MPF automation token position 8 is overlaid with the
@ when a message is queued while NetView is inactive to indicate potentially old messages. If you specify QSSIAT, you must also specify MSGIFAC='QSSIAT' in CNMPSSI.

SSIEXT
Provides the same function as the QSSIAT value. Additionally, with this option, EMCS consoles are obtained when operators or autotasks enter an MVS command, (similar to the SYSTEM value). Unsolicited MVS message traffic flows using the SSI can be queued while NetView or the CNMCSIR task is not active. Solicited (command response) messages are processed using EMCS. This enables you to use the NetView PIPE command for all MVS commands supporting 4-byte console ids and Command and Response Token (CART). If you specify SSIEXT, you must also specify MSGIFAC=SSIEXT in CNMPSSI.

USESSI
Specifies to use the subsystem interface. The task with the load module named CNMCSSIR uses the subsystem interface to route solicited and unsolicited MVS messages, and commands entered from MVS consoles to the NetView program. NetView operators use the consoles defined in the CONSOLxx member, which can be allocated to the subsystem, for issuing system commands from NetView. If you specify USESSI, you must also specify MSGIFAC='USESSI' in CNMPSSI.

NetID
The NetID statement specifies the network ID as a global variable. This statement is coded in CNMSTYLE.

The syntax for the NetID statement is:

```
NetID

NetID=network_id
```

Where:

*network_id*
Indicates a 1–5 character network identifier.

Usage Notes:
* If you set the system symbolic &CNMNETID in member IEASYMxx (SYS1.PARMLIB), the symbol can be used in CNMSTYLE. This is the default.
* The value for NetID is also used for the common CNMSTYLE variable CNMSTYLE.NETID.
* The value specified for NetID is also returned by the netid() REXX function.

NLDM.AMLUTDLY
The NLDM.AMLUTDLY statement enables you to specify the number of seconds the NetView program waits before trying again to access the domain table built by the DSIAMLUT task. This statement is coded in CNMSTYLE.

The syntax for the NLDM.AMLUTDLY statement is:
**NLDM.AMLUTDLY**

Where:

seconds

Specifies the number of seconds from 1–30. The default is 30. If you use a value that is not valid, you receive an error message and the NetView program uses the default value.

Usage Notes:

- If you need to use the session monitor when the DSIAMLUT task is not active, coding a lower value results in faster initialization of the session monitor. However, if the DSIAMLUT task is not active, the session monitor cannot collect data.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.AUTHDOM**

The NLDM.AUTHDOM statement enables you to initialize cross-domain authorization. You can code as many of these statements in CNMSTYLE as needed for your environment.

The syntax for the NLDM.AUTHDOM statement is:

```
NLDM.AUTHDOM.suffix=*

* ANY *

* NONE *

domainid
```

Where:

suffix

The suffix consists of 1–8 characters and must be unique (A,B,C in the example).

NLDM.AUTHDOM.A
NLDM.AUTHDOM.B
NLDM.AUTHDOM.C

*ANY*

Indicates that any operator can establish a cross-domain session (SDOMAIN) with this NetView program. This is the initial setting in CNMSTYLE.

*NONE*

Indicates that no operators are authorized to establish a cross-domain session (SDOMAIN) with this NetView program.

Specifying NONE provides security for both the SDOMAIN and the TRACE commands when using the domainid variable.

domainid

Indicates that any operator using the specified NetView program can establish a cross-domain session (SDOMAIN) with this NetView program.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
**NLDM.AUTHORIZ**

The NLDM.AUTHORIZ statement enables you to specify whether NetView operators in other networks can view session configuration and trace data collected by this NetView program. This data is displayed by scrolling left or right on the Session Configuration panel for a cross-network session. You can code up to 255 statements in CNMSTYLE as needed for your environment.

The syntax for the NLDM.AUTHORIZ statement is:

```
NLDM.AUTHORIZ.suffix=*ANY*,*NONE*,
netid
```

**Where:**

**suffix**

The suffix consists of 1–8 characters and must be unique (A,B,C in the example).

- `NLDM.AUTHORIZ.A`
- `NLDM.AUTHORIZ.A`
- `NLDM.AUTHORIZ.B`

**ANY**

Indicates that NetView operators in all other networks are authorized to view session configuration and trace data collected by this NetView program. This is the initial setting in CNMSTYLE.

**NONE**

Indicates that no NetView operator in another network is authorized to view session configuration and trace data collected by this NetView program.

**Note:** Specifying "NONE" might not provide security in all session configurations.

**netid**

Specifies the network identifier.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

---

**NLDM.CDRMDEF**

The NLDM.CDRMDEF statement sets up CDRMNAME-DOMAIN relationships for other domains. This allows the DST initialization exit for the access method LU task (DSIAMLUT), a Tivoli-supplied initialization exit routine, to build a table of these relationships. The session monitor uses the table for session data retrieval from other domains. You can also use this statement to eliminate session monitor conversation setup attempts by using the RETRY keyword. For a single-domain session monitor configuration, an NLDM.CDRMDEF is not needed.

You code this statement in CNMSTYLE.

The syntax for the NLDM.CDRMDEF statement is:
**NLDM.CDRMDEF**

```
NLDM.CDRMDEF.suffix=cdrmname domainid

---YES---
---NO---
```

**Where:**

**suffix**

The suffix consists of 1–8 characters and must be unique (A,B,C in the example).

- NLDM.CDRMDEF.A
- NLDM.CDRMDEF.B
- NLDM.CDRMDEF.C

**cdrmname**

Specifies the 1–8 character name of a cross-domain SSCP known in this domain as defined to VTAM.

**domainid**

Is the 1–5 character NetView program identifier that names the NetView application in the domain defined by cdrmname.

**YES | NO**

Indicates whether to override the specified NLDM.RETRY value for this particular NLDM.CDRMDEF statement.

- **YES** Indicates that the session monitor attempts to establish an initial conversation with other NetView programs once every 10 minutes.
- **NO** Indicates that the session monitor attempts to establish an initial conversation with other NetView programs only one time.

**Usage Notes:**

- The maximum number of NLDM.CDRMDEF statements that you can code is 65535. If this maximum is exceeded, session monitor initialization fails.
- To implement member changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**Related Statements:** LUC.CNMTARG, DSTINIT, NCCFID, TASK

**NLDM.CDTIME**

The NLDM.CDTIME statement specifies the time-out value for session monitor cross-domain commands and requests. These cross-domain commands can be any session monitor command or request that requires cross-domain communication or cross-domain data. For commands that run in another domain (SDOMAIN was issued), a slightly greater time-out value is used in the home domain in case the other domain issues a cross-domain request for data using CDTIME as the time-out value. This statement is coded in CNMSTYLE.

The syntax for the NLDM.CDTIME statement is:

```
NLDM.CDTIME=seconds
```
**NLDM.CDTIME**

Where:

seconds

Specifies the number of seconds in the range of 1–300. The default is 60 seconds. If you code a value that is not valid, an error message is issued and the default value is used.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.DRDELAY**

The NLDM.DRDELAY specifies the number of seconds to wait for RTM or PIU trace data before recording a session to the session monitor database. This statement is coded in CNMSTYLE.

The syntax for the NLDM.DRDELAY statement is:

`NLDM.DRDELAY=seconds`

Where:

seconds

Specifies the number of seconds in the range of 1–60; values less than 8 should only be used if you are not using the RTM function and you are experiencing a problem with mismatched trace data for a session (session setup PIUs following session end PIUs in the trace for a specific session). The default is 8 seconds. If you specify a value that is not valid, an error message is issued and the NetView program uses the default.

Usage Notes:
- Keep NLDM.DRDELAY to the minimum delay required to avoid backups of data in the session monitor.
- You can omit this statement if you do not have a problem with missing RTM or PIU data, or if you do not use the session monitor TRACE command.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.DSRBO**

The NLDM.DSRBO statement specifies the number of DSRBOs to be associated with the AAUTSKLP task. This statement is coded in CNMSTYLE.

The syntax for the NLDM.DSRBO statement is:

`NLDM.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.

Usage Notes:
- If you change the number, update the STRNO keyword on the BLDVRP macro used to create the LSR pool in CNMSJM01.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.ERCOUNT**

The NLDM.ERCOUNT specifies the number of total explicit routes known to this session monitor. This statement is coded in CNMSTYLE.

The syntax for the NLDM.ERCOUNT statement is:

```
NLDM.ERCOUNT=explicit_routes
```

*Where:*

*explicit_routes*

Specifies the number of explicit routes from all networks known to this session monitor. This value can be 1–999999. This is a tuning value and does not need to be exact. You can approximate this value using the SESSMDIS command (evaluate the TOTAL CURRENT EXPLICIT ROUTES (SARTS) field).

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.FCTIME**

The NLDM.FCTIME statement enables you to specify the time that the session monitor waits for a response to a flow control data request. A flow control data request is sent each time an operator requests a flow control data display. If the session monitor does not receive a response within the specified time limit, the session monitor sends message AAU114I to the authorized receiver and message AAU947I to the operator requesting the display. This statement is coded in CNMSTYLE.

The syntax for the NLDM.FCTIME statement is:

```
NLDM.FCTIME=seconds
```

*Where:*

*seconds*

Specifies the number of seconds in the range of 1–9999. The default is 180.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
The NLDM.KEEPDISC statement enables you to specify the number of discarded PIU trace data records to be kept in virtual storage. This statement is coded in CNMSTYLE.

The syntax for the NLDM.KEEPDISC statement is:

```
NLDM.KEEPDISC=records
```

*Where:*

- `records` specifies the number of records in the range of 1–999. The default is 250.

*Usage Note:* To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

If keep classes are defined, the NLDM.KEEPMEM statement specifies the member name that contains the KCLASS and MAPSESS definition statements. This statement is coded in CNMSTYLE.

The syntax for the NLDM.KEEPMEM statement is:

```
NLDM.KEEPMEM=membername
```

*Where:*

- `membername` indicates the 1–8 character DSIPARM member name for keep class definitions.

*Usage Notes:*

- Omit this statement if you are not using keep classes.
- The NLDM.KEEPMEM statement is commented out in CNMSTYLE as shipped with the NetView product.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

The NLDM.KEEPPIU statement specifies the number of PIU trace data records to be kept in virtual storage for all sessions. This statement is coded in CNMSTYLE.

The syntax for the NLDM.KEEPPIU statement is:

```
NLDM.KEEPPIU=records
```

*Where:*

- `records` specifies the number of records in the range of 1–999. The default is 250.
records

Specifies a number in the range of 0–999. The initial setting in CNMSTYLE is 7.

Usage Notes:

- You can override the KEEPPIU value by defining keep classes.
- You can also change the KEEPPIU value for an individual session by using the KEEP PIU command. For more information on the KEEP command, refer to the NetView online help.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

---

**NLDM.KEEPRTM**

The NLDM.KEEPRTM statement specifies the number of response time collection periods that can be kept in virtual storage for a specific session. This statement is coded in CNMSTYLE.

The syntax for the NLDM.KEEPRTM statement is:

```plaintext
NLDM.KEEPRTM=periods
```

Where:

- **periods**
  
  Specifies a number in the range of 1–999. The default is 10.

Usage Note: You can change RTM keep counts only at initialization.

---

**NLDM.KEEPSESS**

The NLDM.KEEPSESS statement specifies whether DASD session wrapping is used. This statement is coded in CNMSTYLE.

The syntax for the NLDM.KEEPSESS statement is:

```plaintext
NLDM.KEEPSESS=number
```

Where:

- **number**
  
  Specifies the session wrap count. If you specify 0, session wrapping is not used regardless of any KCLASS KEEPSESS values. Also, sessions are not recorded into DGROUPs as defined on a KCLASS statement. This is the default.

  If you specify a **number**, the value is used as the global DASD session wrap count for sessions not mapped by MAPSESS or KCLASS statements and for mapped sessions having no KEEPSESS coded.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
The NLDM.LOG statement specifies whether the NetView program writes records to an external log. The external log can be the system management facilities (SMF) log. This statement is coded in CNMSTYLE.

The syntax for the NLDM.LOG statement is:

```
NLDM.LOG= YES
NO
```

Where:

YES
- Specifies to write records to the external log.

NO
- Specifies that records will not be written to the external log. This is the default.

Usage Notes: NLDM.LOG and NLDM.SESSTATS statements determine what information is written to the external log. Table 9 shows valid combinations for these statements.

Table 9. Combinations of NLDM.LOG and NLDM.SESSTATS

<table>
<thead>
<tr>
<th>If you specify:</th>
<th>The NetView program writes to the external log:</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLDM.LOG=YES</td>
<td>• Response time data (if SAW=YES and RTM=YES)</td>
</tr>
<tr>
<td>NLDM.SESSTATS=YES</td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Availability and accounting data:</td>
</tr>
<tr>
<td></td>
<td>• Session start records, session end records, combined session start-end records</td>
</tr>
<tr>
<td></td>
<td>• Session statistics (PIU counts)</td>
</tr>
<tr>
<td>NLDM.LOG=YES</td>
<td>• Response time data (if SAW=YES and RTM=YES)</td>
</tr>
<tr>
<td>NLDM.SESSTATS=NO</td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Combined session start-end records</td>
</tr>
<tr>
<td>NLDM.LOG=NO</td>
<td>No session monitor data, regardless of the SESSTATS parameter. This is the default.</td>
</tr>
<tr>
<td>NLDM.LOG=YES</td>
<td>• Response time data (if SAW=YES and RTM=YES).</td>
</tr>
<tr>
<td>NLDM.SESSTATS=AVAIL</td>
<td>• Configuration data</td>
</tr>
<tr>
<td></td>
<td>• Availability data (if a KCLASS statement specifies, or defaults to, AVAIL=YES)</td>
</tr>
<tr>
<td></td>
<td>Availability data includes session start records, session end records, and combined session start-end records.</td>
</tr>
</tbody>
</table>

For more information about the record formats that the NetView program writes to the external log, refer to "Tivoli NetView for z/OS Application Programmer’s Guide."
To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE GHB command.

**NLDM.LUCOUNT**

The NLDM.LUCOUNT statement specifies the number of logical units (LUs) in this network. This is a performance tuning value, and does not need to be exact. This statement is coded in CNMSTYLE.

The syntax for the NLDM.LUCOUNT statement is:

```
NLDM.LUCOUNT=number
```

Where:

- **number**
  - Specifies a value in the range of 1–999999.
  - When you assign a value, be sure to include all LUs owned by the local system services control point (SSCP) as well as all LUs owned by other SSCPs that can have session partners owned by the local SSCP.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.MACRF**

The NLDM.MACRF statement specifies the local shared resource (LSR) options. This statement is coded in CNMSTYLE.

The syntax for the NLDM.MACRF statement is:

```
NLDM.MACRF=DRF

LSR

NSR
```

Where:

- **DFR**
  - Extends LSR to defer writing of records. The deferred write (DFR) option defers the writing of a record until the NetView program forces it out because buffer space is needed for a read. This further reduces I/O by minimizing writes.

  Attention: If DFR is coded, do not use the MVS operator CANCEL command or the MVS STOP FORCE command to end the NetView program. Issuing these commands can damage the NetView databases by preventing the final writing of records for which output was deferred. To end the NetView program, use the NetView CLOSE or CLOSE IMMED command.

- **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 default.
**NLDM.MACRF**

`NSR`

Indicates that the data set does not use shared resources.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

---

**NLDM.MAXEND**

The NLDM.MAXEND statement specifies the number of concurrent requests for PIU trace data. This statement is coded in CNMSTYLE.

The syntax for the NLDM.MAXEND statement is:

```
NLDM.MAXEND=number
```

**Where:**

`number`

Specifies a value in the range of 1–999.

**Usage Notes:**

- When the value of `number` is exceeded, you receive message AAU081I. If you receive this message frequently, increase the value. If the specified value is too low, an operator viewing the trace data might not see the most recent PIU data from an active session.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

---

**NLDM.NETID**

The NLDM.NETID statement specifies the name of this network, if you have an interconnected network and have not defined a NETID to VTAM. This statement is coded in CNMSTYLE.

The syntax for the NLDM.NETID statement is:

```
NLDM.NETID=netid
```

**Where:**

`netid`

Specifies the 1–8 character name of this network.

**Usage Notes:**

- The NLDM.NETID statement is commented out in CNMSTYLE as shipped with the NetView product.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
The NLDM.PDDNM statement specifies the session monitor primary data set. This statement is coded in CNMSTYLE.

The syntax for the NLDM.PDDNM statement is:

```
NLDM.PDDNM=name
```

Where:

*name*

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

Usage Notes:

- If necessary, specify the VSAM password in CNMSTPWD.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

The NLDM.PERFMEM statement specifies the DSIPARM member name for performance class definitions. This statement is coded in CNMSTYLE.

The syntax for the NLDM.PERFMEM statement is:

```
NLDM.PERFMEM=membername
```

Where:

*membername*

Specifies the 1–8 character member name for performance class definitions for the response time monitor. If performance classes are defined, this statement specifies the member name that contains the PCLASS and MAPSESS definition statements.

Usage Notes:

- Omit the NLDM.PERFMEM statement if you do not use the NetView program to collect RTM data or if you do not use keep classes.
- This statement is commented out in CNMSTYLE as shipped with the NetView product.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
**NLDM.PIUTNUM**

The NLDM.PIUTNUM statement specifies the number of PIU trace data buffers. This statement is coded in CNMSTYLE.

The syntax for the NLDM.PIUTNUM statement is:

```
NLDM.PIUTNUM=number
```

Where:

`number`

Specifies a value in the range of 2–255. The initial setting in CNMSTYLE is 2.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.PIUTSIZE**

The NLDM.PIUTSIZE specifies the size of the PIU trace data buffers. This statement is coded in CNMSTYLE.

The syntax for the NLDM.PIUTSIZE statement is:

```
NLDM.PIUTSIZE=buffer_size
```

Where:

`buffer_size`

Specifies the buffer size in a range from 2K–32K. K equals 1024. The initial setting in CNMSTYLE is 4K.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

**NLDM.PURGE**

The NLDM.PURGE statement specifies whether additional records are written to the database to increase PURGE and PURGEDB processing speed. This statement is coded in CNMSTYLE.

The syntax for the NLDM.PURGE statement is:

```
NLDM.PURGE=SPEED
```

```
NLDM.PURGE=DASD
```

Where:

`SPEED`

Indicates that additional records are written to the database. This is the default value.
DASD
Indicates that additional records are not written to the database.

Usage Notes:
- Use SPEED if you use NLDM PURGE significantly. Use DASD if you control the database with other means such as KEEPSESS or RESETDB.
- If you change the NLDM.PURGE specification, delete and redefine session monitor databases to avoid mixing session data that was recorded with and without additional records. Failure to do this causes undesirable results with the PURGE and PURGEDB commands. For example, when purging with full wildcards for resource names (as in NLDM PURGE SESSION * * BEFORE date/time or PURGEDB SESS BEFORE date/time), only sessions recorded with additional records are deleted. To delete sessions recorded without additional records, re-issue the PURGE command with the same parameters.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.RETRY
The NLDM.RETRY statement specifies whether additional attempts are made to establish a conversation between this NetView program and the domains specified by NLDM.CDRMDEF statements, if the initial attempt fails. This statement is coded in CNMSTYLE.

The syntax for the NLDM.RETRY statement is:

```
NLDM.RETRY=YES NO
```

Where:
- **YES**
  - Indicates the session monitor attempts to establish an initial conversation with the other domains once every 10 minutes. YES is the default.
- **NO**
  - Indicates the session monitor attempts to establish an initial conversation with the other domains only once.

Usage Notes:
- The RETRY specification on the NLDM.CDRMDEF statement can override this default for specific domains.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.RTDASD
The NLDM.RTDASD statement specifies whether explicit route (ER) data is to be written to the database. This statement is coded in CNMSTYLE.

The syntax for the NLDM.RTDASD statement is:
NLDM.RTDASD

Where:

YES
Indicates that ER data is written to the database. This is the default.

NO
Indicates that ER data is not written to the database.

Usage Notes:

- If a syntax error for the NLDM.RTDASD statement occurs, error message
  AAU096I is issued when the DSILog task is started during NetView
  initialization. The default value is then used.
- To implement definition changes while the NetView program is active, change
  the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.RTM

The NLDM.RTM statement enables the NetView response time monitor (RTM)
function. This statement is coded in CNMSTYLE.

The syntax for the NLDM.RTM statement is:

Where:

YES
Indicates that the NetView response time monitor function is enabled.

NO
Indicates that the NetView response time monitor function is not enabled. This
is the default value.

Usage Note: To implement definition changes while the NetView program is
active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.RTMDISP

The NLDM.RTMDISP statement specifies whether you can display response times
at your workstation in the operator information area. This statement is coded in
CNMSTYLE.

The syntax for the NLDM.RTMDISP statement is:

Where:
YES
Indicates that you can display response times. This is the default.

NO
Indicates that you cannot display response times.

Usage Notes:
• The NLDM.RTMDISP specification overrides the value specified in the hardware configuration.
• The DSPLYLOC operand of the PCLASS statement overrides the NLDM.RTMDISP specification.
• To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.SAW
The NLDM.SAW statement specifies whether session awareness (SAW) data should be collected. This statement is coded in CNMSTYLE.

The syntax for the NLDM.SAW statement is:

```plaintext
NLDM.SAW= YES
```

Where:
YES
Indicates that SAW data is collected, beginning at NetView program initialization. This is the default.

NO
Indicates that no session awareness data is collected until the ENABLE command is entered.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.SAWNUM
The NLDM.SAWNUM statement specifies the number of session awareness (SAW) data buffers. This statement is coded in CNMSTYLE.

The syntax for the NLDM.SAWNUM statement is:

```plaintext
NLDM.SAWNUM=number
```

Where:

number
Specifies a value in the range of 2–255. The initial setting in CNMSTYLE is 2.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.
NLDM.SAWSIZE

The NLDM.SAWSIZE statement specifies the size of session awareness (SAW) data buffers. This statement is coded in CNMSTYLE.

The syntax for the NLDM.SAWSIZE statement is:

```
NLDM.SAWSIZE=buffer_size
```

Where:

- `buffer_size` specifies the buffer size in a range from 2K–32K. K equals 1024. The initial setting in CNMSTYLE is 4K.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.SDDNM

The NLDM.SDDNM statement specifies session monitor secondary data set. This statement is coded in CNMSTYLE.

The syntax for the NLDM.SDDNM statement is:

```
NLDM.SDDNM=name
```

Where:

- `name` indicates the 1–8 character DD name of the secondary data set to be used by VSAM services. The default is AAUVSSL.

Usage Notes:
- If necessary, specify the VSAM password in CNMSTPWD.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.SESSTATS

The NLDM.SESSTATS statement specifies whether the network accounting and availability functions are active. This statement is coded in CNMSTYLE.

The syntax for the NLDM.SESSTATS statement is:

```
NLDM.SESSTATS=NO AVAIL YES
```

Where:
NO
Specifies that network accounting and availability functions are not active. This is the default.

AVAIL
Specifies that only the availability function is active.

YES
Specifies that both the network accounting and availability functions are active.

Usage Notes:
- NLDM.LOG and NLDM.SESSTATS statements determine what information is written to the external log. Table 9 on page 92 shows valid combinations for these statements.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.TRACEGW
The NLDM.TRACEGW statement specifies whether the NetView program should activate gateway tracing at NetView program initialization or NCP activation. This statement is coded in CNMSTYLE.

The syntax for the NLDM.TRACEGW statement is:

```
NLDM.TRACEGW= NO
```

Where:

NO
Indicates that the NetView program does not activate gateway tracing for all NCPs for which session awareness data is received.

YES
Indicates that the NetView program activates gateway tracing for all NCPs for which session awareness data is received. This is the default.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

NLDM.TRACELU
The NLDM.TRACELU statement specifies whether the NetView program should start tracing LU information at NetView program initialization. This statement is coded in CNMSTYLE.

The syntax for the NLDM.TRACELU statement is:

```
NLDM.TRACELU= NO
```

Where:
### NLDM.TRACELU

- **NO**
  - Indicates that no LU tracing occurs until the TRACE START command is entered. This is the default value.

- **YES**
  - Indicates that LU-LU sessions are traced beginning at NetView program initialization.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

### NLDM.TRACEESC

The NLDM.TRACEESC statement specifies whether the NetView program should start tracing SSCP information at NetView program initialization. This statement is coded in CNMSTYLE.

The syntax for the NLDM.TRACEESC statement is:

```
NLDM.TRACEESC= NO
```

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NLDM command.

### NMCstatus

The NMCstatus statement defines a data set that contains any errors found in the GRAPHICS policy file. This statement is coded in CNMSTYLE.

The syntax for the NMCstatus statement is:

```
NMCstatus.errorDSN=data_set_name
```

**Where:**

- **data_set_name**
  - Data set name. You can specify *NONE* if no data set is needed.

**Usage Note:**
- Many of the messages are greater than 80 characters in length so if a FIXED 80 file is specified, most messages are truncated.
NMCSTATUS (Control File Entry)

The NMCSTATUS control file entry, or policy definition, enables you to define time schedules for resources in NMC views. With these schedules, policy can be applied to NMC views to specify when:

- The displayable status of one or more resources in a view is disabled at the NMC console.
- One or more resources in a view is suspended from aggregation.
- The displayable status of one or more resources in a view is disabled at the NMC console and the same resources are suspended from aggregation.

Resources can be grouped by a RODM class name, a BLDVIEWS definition, or a RODM Collection Manager definition.

Creating and Loading a Policy File

NMCSTATUS policy definitions are defined in a policy file. The policy file must be a DSIPARM member. To have your NMCSTATUS policy definitions loaded in the Policy Repository when NetView is initialized, define your policy file in CNMSTYLE with the following statement:

```
POLICY.GRAPHICS = DUIPOLCY
```

DUIPOLCY is shipped in DSIPARM as a sample NMCSTATUS policy file. You can use this file name or create your own DSIPARM member. Refer to the comments in DSIPARM member CNMSTYLE for more information on policy definitions.

NMCSTATUS Policy Autotask

If you specified TOWER GRAPHICS in CNMSTYLE, the NMCSTATUS policy autotask is started at NetView initialization. Statements are coded in DSIPARM members CNMSTYLE and CNMSTASK to facilitate the starting of the task and in DSIPARM member DSITBL01 to begin processing the NMCSTATUS policy definitions. It is possible to change the name of the DUIFPOLI autotask to your own 8-character autotask name. In DSIPARM member CNMSTYLE the statement

```
function.autotask.NMCpolicy = DUIFPOLI
```

results in the creation of common global variable

```
CNMSTYLE.AUTO.NMCpolicy = DUIFPOLI
```

By editing DUIFPOLI in this statement, you can change the name of the autotask. Global CNMSTYLE.AUTO.NMCpolicy is referenced in DSIPARM members CNMSTASK and DSITBL01; do not change these statements.

The statement RODMname = &CNMRODM defined in DSIPARM member CNMSTYLE must contain the RODM nickname. If it does not, processing of NMCSTATUS policy definitions fails.

Because NMCSTATUS processing is handled by an autotask, all messages issued by commands NMCPINIT and NMCPTEST appear only in the NetView log. If the NMCPINIT or NMCPTEST command is run at a NetView OST, the first thing the command does is EXCMD itself to the autotask.
Processing a Policy File

If the NMCSTATUS policy file is successfully read into storage at NetView initialization, the following message is issued:
EZL110I NVPOLICY BEING USED FOR THE CONFIGURATION TABLE

This message is automated in DSIPARM member DSITBL01 to execute command NMCPINIT on autotask DUIFPOLI. The NMCPINIT command reads the policy file specified by the POLICY.GRAPHICS statement and determines if the NMCSTATUS policy definitions are valid.

If a DUIPOLICY file is found, the following message is issued indicating NMCSTATUS policy processing has begun:
DUI275I BEGIN PROCESSING NMCSTATUS POLICY DEFINITIONS DEFINED IN MEMBER DUIPOLICY

Both the NMCPINIT and NMCPTEST commands error check the NMCSTATUS policy definitions. A multiline message is issued containing any errors found in the policy file. The following is an example of an error free policy file:
DUI250I -------------------------------------------
DUI251I BEGIN ERROR CHECKING FOR MEMBER DUIPOLICY
DUI250I -------------------------------------------
DUI261I NO ERRORS WERE FOUND IN MEMBER DUIPOLICY
DUI250I -------------------------------------------
DUI252I END ERROR CHECKING FOR MEMBER DUIPOLICY
DUI250I -------------------------------------------

In addition to the log, the messages can be written to a data set. Edit DSIPARM member CNMSTYLE and replace *NONE* with the name of your allocated data set:
NMCstatus.errorDSN = *NONE*

Note: Many of the messages are greater than 80 characters in length so if a FIXED 80 file is specified, most messages are truncated.

If no errors were found, the autotask creates CHRON timers to specify the beginning and end of each policy window. Each policy definition generates two CHRON timers. One timer indicates the time the policy begins and the second timer indicates the time the policy ends. Timers created by the NMCSTATUS policy autotask are prefixed with the characters NMC, for example NMC1. To see all timers created by the DUIFPOLI autotask enter:
TIMER NMC

where NMC is the filter. For information on setting up security for these timers, refer to the Tivoli NetView for z/OS Security Reference.

If all timers are set successfully, the following multiline message is issued:
DUI280I --------------------------------------------------------------------------
DUI281I BEGIN SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLICY
DUI280I --------------------------------------------------------------------------
DUI283I ALL TIMERS WERE SET SUCCESSFULLY
DUI280I --------------------------------------------------------------------------
DUI281I END SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLICY
DUI280I --------------------------------------------------------------------------

If a timer could not be set, messages DUI253E and DUI284E are issued as part of the above multiline message, followed by any CHRON error messages. For
example, if keyword CALENDARDAY=(PAYDAY) is defined for policy definition POLICY1, but PAYDAY is not defined in DSIPARM member DSISCHED, the following multiline message is issued:

DUI280I ----------------------------------------------------------------------  
DUI281I BEGIN SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLCY  
DUI280I ----------------------------------------------------------------------  
DUI253E POLICY1: UNABLE TO CREATE BEGINNING TIMER FOR THIS POLICY DEFINITION  
IN MEMBER DUIPOLCY  
DUI284E COMMAND EZLETAPI FAILED WITH RETURN CODE 24 IN MODULE DUIFTIMR. THE NEXT  
1 MESSAGE(S) MAY HELP DETERMINE THE ERROR  
CMN249E DSISCHED : NO DATA FOUND FOR 'PAYDAY'  
DUI273E POLICY1: UNABLE TO CREATE ENDING TIMER FOR THIS POLICY DEFINITION IN  
MEMBER DUIPOLCY  
DUI284E COMMAND EZLETAPI FAILED WITH RETURN CODE 24 IN MODULE DUIFTIMR. THE NEXT  
1 MESSAGE(S) MAY HELP DETERMINE THE ERROR  
CMN249E DSISCHED : NO DATA FOUND FOR 'PAYDAY'  
DUI280I ----------------------------------------------------------------------  
DUI282I END SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLCY  
DUI280I ----------------------------------------------------------------------  

Your timers may not always be set successfully. When the NMCSTATUS policy definitions are error checked, test CHRON timers are set to make sure the values you specified for keywords TIME, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY, and ECALENDARDAY are valid. Any errors found are issued as part of the multiline message beginning with DUI251I and ending with DUI252I.

When the timers are set NMCSTATUS processing is complete and the following message is issued:  
DUI276I END PROCESSING NMCSTATUS POLICY DEFINITIONS DEFINED IN MEMBER DUIPOLCY,  
RETURN CODE 0  

Errors Found in a Policy File

The following is an example of the multiline message when the policies contain an error.

DUI250I ----------------------------------------------------------------------  
DUI251I BEGIN ERROR CHECKING FOR MEMBER DUIPOLCY  
DUI250I ----------------------------------------------------------------------  
DUI257E POLICY1: TIME IS A REQUIRED KEYWORD MISSING FOR THIS POLICY DEFINITION  
IN MEMBER DUIPOLCY  
DUI250I ----------------------------------------------------------------------  
DUI252I END ERROR CHECKING FOR MEMBER DUIPOLCY  
DUI250I ----------------------------------------------------------------------  

Until all errors are resolved, actions based on these policies, such as, disabling resource status changes or suspending resources from aggregation, will not occur at the NMC console.

To update your policy file without recycling NetView:

• Update the DSIPARM member defined by POLICY.GRAPHICS in CNMSTYLE  
• Issue the POLICY REQ=LOAD command  
• Automation on message EZL110I executes command NMCPINIT to begin NMCSTATUS policy processing.
NMCSTATUS

To begin NMCSTATUS processing without reloading a policy file, issue the NMCPINIT command. The NMCPINIT command processes NMCSTATUS policy definitions currently loaded in the Policy Repository.

Test a Policy File

It is possible to test a policy file without starting GMFHS, or RODM, or signing onto NMC. The NMCPTEST command allows you to error check NMCSTATUS policy definitions currently loaded in the Policy Repository.

To test your policy file at NetView initialization:

- Update the DSIPARM member defined by POLICY.GRAPHICS in CNMSTYLE
- Update DSIPARM member DSITBL01 such that automation on message EZL110I executes command NMCPTEST, rather than command NMCPINIT, to begin NMCSTATUS policy error checking.
- If errors are found, update the DSIPARM member defined by the POLICY.GRAPHICS statement. Instead of recycling NetView to reload the policy, you can issue the POLICY REQ=LOAD command which via automation on message EZL110I executes command NMCPTEST to error check the NMCSTATUS policies.

The following is the syntax for NMCSTATUS:

```
++NMCSTATUS=policy_definition++
  CLASS=(class_name)
  RESOURCE=(drname)
  MYNAME=(myname)
  BLDVIEWSPEC=(type,definition)
  COLLECTIONSPEC=(type,definition)

++STOPUPDATE=YES
++TIME=(starttime,endtime)
++SUSPENDAGG=YES

++DAYOFWEEK=ALL
++DAYOFWEEK=(dayname)
  1ST
  2ND
  3RD
  4TH
  5TH
  LAST
  LAST -n

++DAYOFMONTH=ALL
++DAYOFMONTH=(dayofmonthnumber)
  LAST
  LAST -n
```
Where:

**NMCSTATUS**

Required for each policy definition. Must be entered in uppercase.

**policy_definition**

The name of your policy definition. The *policy_definition* must be unique for each NMCSTATUS definition, is case sensitive, and can be from 1 to 32 characters with no embedded blanks, commas, or quotes. If you are using AON commands, such as, DSPCFG, then ENTRY=NMCSTATUS and TYPE=policy_definition.
CLASS=(class_name)
One and only one of the following keywords is required: CLASS, BLDVIEWSSPEC, or COLLECTIONSPEC. If CLASS is specified, class_name must be a valid RODM class name and is case sensitive. No wildcarding is allowed. RODM is case sensitive. Only one class is allowed for each NMCSTATUS policy definition, for example:
CLASS=(GMFHS_Managed_Real_Objects_Class)

The maximum length of class_name is 64 characters. For more information on class names, refer to the MyName field in the Tivoli NetView for z/OS Data Model Reference.

BLDVIEWSSPEC=(type_definition)
One and only one of the following keywords is required: CLASS, BLDVIEWSSPEC, or COLLECTIONSPEC. The BLDVIEWSSPEC keyword must specify a type and a definition:

- If type=QSAMDSN, then definition is a fully-qualified data set name. The specification is translated into a PIPE command. For example:
  BLDVIEWSSPEC=(QSAMDSN,USER.INIT(F1))

generates a PIPE command 'PIPE QSAM (DSN) USER.INIT(F1)' to read the BLDVIEWS specification. FILE1 should contain only one BLDVIEWS specification and must be allocated before the policies are processed.

- If type=QSAMDD, then definition is an allocated data definition name. This specification is translated into a PIPE command. For example:
  BLDVIEWSSPEC=(QSAMDD,ALLOCDDD)

generates a PIPE command 'PIPE QSAM (DD) ALLOCDDD' to read the BLDVIEWS specification. ALLOCDDD is the 'FILE' value from the ALLOCATE command, such as:
  ALLOCATE FILE(ALLOCDDD) DATASET(USER.INIT(F1)) SHR

and must be issued to create the data definition file before the policy is active. Refer to HELP PIPE QSAM for more information. FILE1 should contain only one BLDVIEWS specification.

The maximum length of definition is 61 characters if type=QSAMDSN, and 62 characters if type=QSAMDD. This allows for the value to start in column 2 and be enclosed in parenthesis. For example:

BLDVIEWSSPEC=(QSAMDSN,fully_qualified_dataset_name)

COLLECTIONSPEC=(type_definition)
One and only one of the following keywords is required: CLASS, BLDVIEWSSPEC, or COLLECTIONSPEC. The COLLECTIONSPEC keyword must specify a type and definition:

- If type=QSAMDSN, then definition is a fully-qualified data set name. The specification is translated into a PIPE command. For example:
  COLLECTIONSPEC=(QSAMDSN,USER.INIT(F1))

generates a PIPE command 'PIPE QSAM (DSN) USER.INIT(F1)' to read the RODM Collection Manager (RCM) specification. FILE1 should contain only one RCM specification and must be allocated before the policies are processed.
• If type=QSAMDD, then definition is an allocated data definition name. This specification is translated into a PIPE command. For example:

```
COLLECTIONSPEC=(QSAMDD,ALLOCDDD)
```

generates a PIPE command `PIPE QSAM (DD) ALLOCDDD` to read the RCM specification. ALLOCDDD is the 'FILE' value from the ALLOCATE command, such as:

```
ALLOCATE FILE(ALLOCDDD) DATASET(USER.INIT(FILE1)) SHR
```

and must be issued to create the data definition file before the policy is active. Refer to HELP PIPE QSAM for more information. FILE1 should contain only one RCM specification.

The maximum length of definition is 61 characters if type=QSAMDSN, and 62 characters if type=QSAMDD. This allows for the value to start in column 2 and be enclosed in parenthesis. For example:

```
COLLECTIONSPEC=
  (QSAMDSN,definition)
```

**TIME=starttime,endtime**

TIME is a required keyword where:

• **starttime**
  Defines the starting time when actions are applied to resources in NMC views. The actions are defined in the policy definition with the SUSPENDAGG and STOPUPDATE keywords. The resources are defined in the policy definition with the CLASS, BLDVIEWSSPEC, and COLLECTIONSPEC keywords. The time must be specified using the 24-hour clock in hh.mm.ss format. Values are in the range of 00.00.00 to 23.59.59.

• **endtime**
  Defines the ending time when actions are applied to resources in NMC views. The actions are defined in the policy definition with the SUSPENDAGG or STOPUPDATE keywords. The resources are defined in the policy definition with the CLASS, BLDVIEWSSPEC, or COLLECTIONSPEC keywords. The time must be specified using the 24-hour clock in the hh.mm.ss format. Values are in the range of 00.00.00 to 23.59.59.

**DAYOFWEEK**

Specifies the name of the weekday the policy is activated.

DAYOFWEEK is an optional keyword. The default is DAYOFWEEK=ALL. For a policy to be activated on a particular day, DAYOFWEEK, DAYOFMONTH, and CALENDARDAY must be evaluated collectively. For example, if

```
DAYOFWEEK=(FRI)
DAYOFMONTH=ALL
CALENDARDAY=(HOLIDAY)
```

is specified, the policy is activated only on holidays that are on Friday. To specify all days issue:

```
DAYOFWEEK=ALL
```

All other values must be specified within parenthesis, for example:
DAYOFWEEK= (MON) or DAYOFWEEK= (MON, TUE)

Valid values are:
- SUN
- MON
- TUE
- WED
- THU
- FRI
- SAT
- WEEKDAY
- WEEKEND

Specifying NOT to omit selected days reduces a longer list of days to be included. For example, instead of specifying:
DAYOFWEEK= (TUE, WED, THU, FRI)

you can achieve the same result by specifying:
DAYOFWEEK= (NOT MON, WEEKEND)

and the policy would be active on Tuesdays through Fridays. You can schedule a policy to be active on certain occurrences of that day within a month, for example:
DAYOFWEEK= (MON(1ST, 3RD), FRI(LAST))

results in an active policy only on the first and third Monday, and the last Friday of the month. Unsigning the LAST or LAST-n prevents having to consider the number of a specific weekday within that month. Valid values are:
- 1ST
- 2ND
- 3RD
- 4TH
- 5TH
- LAST
- LAST-1
- LAST-2
- LAST-3
- LAST-4

The maximum length of value for the DAYOFWEEK keyword is 69 characters. This allows for the value to start in column two and be enclosed in parentheses.

DAYOFMONTH
Specifies the number of the day within the month the policy is activated.

DAYOFMONTH is an optional keyword. The default is DAYOFMONTH=ALL. For a policy to be activated on a particular day, DAYOFWEEK, DAYOFMONTH, and CALENDARDAY must be evaluated collectively. For example, if
DAYOFWEEK=ALL
DAYOFMONTH=(15)
CALENDARDAY=(HOLIDAY)

are specified, the policy is activated only on holidays that are on the 15th
day of the month. To specify all days issue:
DAYOFMONTH=ALL

All other values must be specified within parentheses. Valid values are:
• In the range of 1 to 31
• LAST
• In the range of LAST-1 through LAST-30
• NOT

Specifying NOT to omit selected days reduces a longer list of days to be
included. For example, instead of specifying:
DAYOFMONTH=(5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30)

you can achieve the same result by specifying:
DAYOFMONTH=(NOT 1,2,3,4,31)

and the policy is not active on the first, second, third, forth, and thirty-first
day of the month. Specifying LAST or LAST -n eliminates having to
consider the number of days within the month.

The maximum length of value for the DAYOFMONTH keyword is 69
characters. This allows for the value to start in column 2 and be enclosed
in parenthesis.

CALENDARDAY
Specifies the name of a key as defined in DSIPARM member DSISCHED
when the policy is activated.

CALENDARDAY is an optional keyword. The default is
CALENDARDAY=ALL. For a policy to be activated on a particular day,
DAYOFWEEK, DAYOFMONTH, and CALENDARDAY must be evaluated
collectively. For example, if
DAYOFWEEK=ALL
DAYOFMONTH=(15,30)
CALENDARDAY=(PAYDAY)

are specified, the policy is activated only on days defined as PAYDAY that
are also on the 15th or 30th day of the month. To specify all days issue:
CALENDARDAY=ALL

All other values must be specified within parentheses. If NOT is specified,
the policy is not active on the specified days. You can enter up to 1,000
unique keys in the list. If you exceed this limit, message DSI656I is issued.

The maximum length of value for the CALENDARDAY keyword is 69
characters. This allows for the value to start in column two and be
enclosed in parenthesis.

If you add a key to DSISCHED for today, you must issue a CHRON
REFRESH command to make the calendar active. Otherwise, any changes
made are not effective after the next midnight.
EDAYOFWEEK

Specifies the name of the weekday the policy is deactivated.
EDAYOFWEEK is an optional keyword but if specified, DAYOFWEEK is a
required keyword. If both DAYOFWEEK and EDAYOFWEEK are specified,
DAYOFWEEK is used in the beginning timer to indicate when the policy is
activated and EDAYOFWEEK is used in the ending timer to indicate when
the policy is deactivated. If EDAYOFWEEK is not specified, the ending
timer defaults to the values used for the beginning timer.

The same values are valid for EDAYOFWEEK as DAYOFWEEK.

For a policy to be deactivated on a particular day, EDAYOFWEEK,
EDAYOFMONTH, and ECALENDARDAY must be evaluated collectively.
For example, if

DAYOFWEEK=(FRI)
EDAYOFWEEK=(SAT)
DAYOFMONTH=ALL
CALENDARDAY=(HOLIDAY)

is specified, the policy is activated on holidays that are on Friday but is
deactivated only on holidays that are on Saturday. The ending timer
defaults to the values for DAYOFMONTH and CALENDARDAY.

EDAYOFMONTH

Specifies the number of the day within the month the policy is deactivated.
EDAYOFMONTH is an optional keyword but if specified, DAYOFMONTH
is a required keyword. If both DAYOFMONTH and EDAYOFMONTH are
specified, DAYOFMONTH is used in the beginning timer to indicate when
the policy is activated and EDAYOFMONTH is used in the ending timer to
indicate when the policy is deactivated. If EDAYOFMONTH is not
specified, the ending timer defaults to the values used for the beginning
timer.

The same values are valid for EDAYOFMONTH as DAYOFMONTH.

For a policy to be deactivated on a particular day, EDAYOFWEEK,
EDAYOFMONTH, and ECALENDARDAY must be evaluated collectively.
For example if

DAYOFWEEK=ALL
DAYOFMONTH= (15)
EDAYOFMONTH= (16)
CALENDARDAY= (HOLIDAY)

is specified, the policy is activated on holidays that are on the 15th day of
the month but is deactivated only on holidays that are on the 16th day of
the month. The ending timer defaults to the values for DAYOFMONTH
and CALENDARDAY.

ECALENDARDAY

Specifies the name of a key, as defined in DSIPARM member DSISCHED,
when the policy is deactivated. ECALENDARDAY is an optional keyword
but if specified, CALENDARDAY is a required keyword. If both
CALENDARDAY and ECALENDARDAY are specified, CALENDARDAY is
used in the beginning timer to indicate when the policy is activated and
CALENDARDAY is used in the ending timer to indicate when the policy is
deactivated. If ECALENDARDAY is not specified, the ending timer
defaults to the values used for the beginning timer.

The same values are valid for ECALENDARDAY as CALENDARDAY.
If you add a key to DSISCHED for today, you must issue a CHRON REFRESH command to make the calendar active. Otherwise, any changes made are not effective after the next midnight.

For a policy to be deactivated on a particular day, EDAYOFWEEK, EDAYOFMONTH and ECALENDARDAY must be evaluated collectively. For example if

```
DAYOFWEEK=ALL
DAYOFMONTH=(25)
CALENDARDAY=(HOLIDAY)
ECALENDARDAY=(DAY_AFTER_HOLIDAY)
```

is specified, the policy is activated on holidays that are on the 25th day of the month but is deactivated only on days after holidays that are also on the 25th day of the month. The ending timer defaults to the values for DAYOFMONTH and CALENDARDAY.

**SUSPENDAGG=YES/NO**

- **YES**  Suspend resources from aggregation.
- **NO**  Do not suspend resources from aggregation. NO is the default.

One of the following keyword/value pairs must be specified for each NMCSTATUS policy definition: SUSPENDAGG=YES or STOPUPDATE=YES.

**STOPUPDATE=YES/NO**

- **YES**  Disable status updates for resources.
- **NO**  Allow status updates for resources. No is the default.

One of the following keyword/value pairs must be specified for each NMCSTATUS policy definition: SUSPENDAGG=YES or STOPUPDATE=YES.

**RESOURCE=(drname)**

The DisplayResourceName of the specific resource or group of resources to which these values apply. You can use the wildcard character asterisk (*) or question mark (?) to specify groups of resources. The MYNAME and RESOURCE keywords cannot both be specified in the same NMCSTATUS policy definition. The RESOURCE keyword can only be specified if the CLASS keyword is specified.

The maximum length of drname is 69 characters. This allows for the value to start in column two and be enclosed in parentheses.

**MYNAME=(myname)**

The MyName of the specific resource or group of resources to which these values apply. You can use the wildcard character asterisk (*) or question mark (?) to specify groups of resources. The MYNAME and RESOURCE keywords cannot both be specified in the same NMCSTATUS policy definition. The MYNAME keyword can only be specified if the CLASS keyword is specified.

The maximum length of drname is 69 characters. This allows for the value to start in column two and be enclosed in parentheses.

**NMCSTATUS Policy Definition Format**

- NMCSTATUS must be specified in upper case and begin in column one.
Continuation of NMCSTATUS Policy Definitions

Continued lines must begin after column one, for example:

RESOURCE=
(ResourceNameThatIsSixtyNineCharactersLong)

enables you to utilize all 69 characters. Wildcard characters can also be used in these values. Columns 73 to 80 are restricted from use.

Values within parentheses cannot be continued to the next line. This, in addition to the restricted use of columns 73 to 80, imposes a length restriction on values for the RESOURCE, MYNAME, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY, ECALENDARDAY and CLASS keywords.

Case Rules for NMCSTATUS Policy Definitions

Keywords can be specified in mixed case but are interpreted by the NMCPINIT and NMCPTEST commands as upper case.

Values for the following keywords can be specified in mixed case but are interpreted by the NMCPINIT and NMCPTEST commands as upper case: BLDVIEWSSPEC, COLLECTIONSPEC, SUSPENDAGG, STOPUPDATE, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY, ECALENDARDAY, and CLASS.

The BLDVIEWS and RODM Collection Manager definitions contained in data sets or data definition files and specified on keywords BLDVIEWSSPEC and COLLECTIONSPEC are case sensitive and are not changed by the NMCPINIT or NMCPTEST commands.

Values for the following keywords are case sensitive and are not changed by the NMCPINIT or NMCPTEST commands: CLASS, RESOURCE, and MYNAME.

Required Keywords for NMCSTATUS Policy Definitions

One and only one of the following keywords is required: CLASS, BLDVIEWSSPEC, or COLLECTIONSPEC

TIME is a required keyword

At least one of the following keywords is required: SUSPENDAGG=YES or STOPUPDATE=YES
RESOURCE is an optional keyword, but if specified, then CLASS is a required keyword and MYNAME is an invalid keyword.

MYNAME is an optional keyword, but if specified, then CLASS is a required keyword and RESOURCE is an invalid keyword.

DAYOFWEEK is an optional keyword.

DAYOFMONTH is an optional keyword.

CALENDARDAY is an optional keyword.

If DAYOFWEEK, DAYOFMONTH, and CALENDARDAY are not specified, DAYOFWEEK defaults to DAYOFWEEK=ALL

EDAYOFWEEK is an optional keyword, but if specified, DAYOFWEEK is a required keyword.

EDAYOFMONTH is an optional keyword, but if specified, DAYOFMONTH is a required keyword.

ECALENDARDAY is an optional keyword, but if specified, CALENDARDAY is a required keyword.

Example NMCSTATUS Policy Definitions

Example 1

The following example illustrates how to stop status updates at NMC for a specific resource.

NMCSTATUS POLICY1
CLASS=(GMFHS_Managed_Real_Objects_Class)
RESOURCE=(DECNET.RALV4)
TIME=(08.00.00,17.00.00)
DAYOFWEEK=(SAT)
STOPUPDATE=YES

or

NMCSTATUS POLICY1
COLLECTIONSPEC=(QSAMDSN,USER.INIT(CSFILE1))
TIME=(08.00.00,17.00.00)
DAYOFWEEK=(SAT)
STOPUPDATE=YES

where CSFILE1 contains this RODM Collection Manager definition
GMFHS_Managed_Real_Objects_Class_MyName_DECNET.RALV4_.EQ.

Example 2

The following example illustrates how to stop status updates at NMC and suspend aggregation for all resources in a class.

NMCSTATUS POLICY2
CLASS=(GMFHS_Managed_Real_Objects_Class)
TIME=(08.00.00,12.00.00)
DAYOFMONTH=(1,15)
STOPUPDATE=YES
SUSPENDAGG=YES

Example 3

The following example illustrates how to suspend a group of resources defined by a BLDVIEWS specification from aggregation on a defined day. PAYDAY is defined in DSIPARM member DSISCHED.
Example 4

The following example illustrates how to stop updates at NMC for a group of resources defined by a BLDVIEWS specification. The updates are stopped on May 1 and started again on June 1. MAY 1 and JUNE 1 are defined in DSIPARM member DSISCHED.

Example 5

The following example illustrates how to stop updates at NMC for a group of resources defined by a BLDVIEWS specification. The updates are stopped at 10pm on Sunday and started again at 6am on Monday.

NMCSTATUS Processing When Beginning Timer Pops

When a timer pops to indicate the beginning of a policy window, an internal command is run on the NMCSTATUS policy autotask to create a RODM object representing the policy. If an object of the same name exists in RODM, the object is deleted and created again for this policy definition. An object cannot be created if:

- You specified the CLASS keyword but the class_name does not exist in RODM.
- You specified a BLDVIEWSSPEC or COLLECTIONSPEC keyword and the data set name or data definition file has been deleted or deallocated since the CHRON timer was initially set.
- You specified the BLDVIEWSSPEC keyword but your BLDVIEWS definition was in error and could not be translated to a RODM Collection Manager definition.

For additional information about the RODM object representing the policy definition and how this information is displayed in an NMC view, refer to Applying Policy to Views in the Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide.

NMCSTATUS Processing When Ending Timer Pops

When a timer pops to indicate the end of a policy window, an internal command is run on the NMCSTATUS policy autotask to delete the RODM object representing the policy.
For additional information about the RODM object representing the policy definition and how this information is displayed in an NMC view, refer to Applying Policy to Views in the Tivoli NetView for z/OS Resource Object Data Manager and GMEHS Programmer’s Guide.

**NOPREFIX**

The NOPREFIX statement prevents prefix information from being displayed in front of the text of a message. It also suppresses the column heading line. The sample member containing the NOPREFIX statement is CNMSCNFT. The NOPREFIX statement overrides PREFIX statements that have been defined.

The syntax for the NOPREFIX statement is:

```
NOPREFIX
```

**Usage Note:** To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

**Related Statements:** ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

---

**NORMAL**

The NORMAL statement enables you to change the color of normal messages that are displayed on the command facility panel. The NORMAL statement also defines the color and highlight attributes of the message area of the command facility panel after the screen is erased using the CLEAR command. The sample member containing the NORMAL statement is CNMSCNFT. Code the NORMAL statement only once.

The syntax for the NORMAL statement is:

```
NORMAL
```

**Where:**

*color*

Defines foreground color for normal messages. The foreground color must be specified before the background color.

*attribute*

Defines alarm, intensity, and highlight attributes for normal messages. You can specify attributes only once for the NORMAL statement.
NORMAL

ON
Makes the color following ON apply to the background of the normal messages. This is a required keyword if you specify only a background color.

colorb
Defines background color for normal messages.

Usage Notes:
• To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
• The NetView-supplied defaults for the NORMAL statement are:
  – Turquoise foreground
  – Black background
  – Normal highlighting
  – Low intensity
  – Alarm off
• The color operands are:
  BLACK The color black
  BLUE The color blue
  RED The color red
  PINK The color pink
  GREEN The color green
  TURQ The color turquoise
  YELLOW The color yellow
  WHITE The color white
• The highlight operands are:
  BLINK The characters blink.
  REV The characters change to reverse video.
  UND The characters are underlined.
  NRM Normal attributes are used.
• The intensity operands are:
  HIGH The characters have high intensity.
  LOW The characters have low intensity.
• The alarm operand is:
  BEEP When a message is shown, an audible alarm sounds.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPI, TITLESTAT, TITLETIME

NORMQMAX

The NORMQMAX statement defines the number of normal messages retained by the command facility for display at a later time (for example, messages kept while you are using another panel or while the screen is locked). When the NORMQMAX limit is exceeded, the NetView program automates and logs the message, if required, and then discards the message without interrupting you. When you roll to the command facility, message DSI593A indicates how many messages you did not see. The NetView program discards the oldest messages until the number of messages remaining is half of the NORMQMAX value. Discarding is done only when NORMQMAX is exceeded. The sample member containing the NORMQMAX statement is CNMSCNFT.
Note: The NORMQMAX value also applies to hardcopy printers and to OST-NNT cross-domain sessions. Hardcopy printers can get backlogged because they are slow or because they are out of paper. An OST-NNT session can get backlogged because message traffic over the session exceeds the session send rate.

The syntax of the NORMQMAX statement is:

```
NORMQMAX
```

Where:

- **3000|nnnnnnnnnn**
  - Specifies the maximum number of normal messages to be kept. If you specify zero, the NetView program does not use a limit. If you specify a value in the range of 1–100, the NetView program rounds the value to 100. If you specify a value greater than 100, the NetView program uses the actual value. The maximum value is 2,147,483,647. The default is 3000.

Usage Notes:

- Specifying too high a value for NORMQMAX can result in short-of-storage conditions.
- To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOID, TITLESTAT, TITLETIME

### NPDA.ALCACHE

The NPDA.ALCACHE statement specifies the maximum size of your alert cache. This statement is coded in CNMSTYLE.

The syntax for the NPDA.ALCACHE statement is:

```
NPDA.ALCACHE
```

Where:

- **10|maxnumber**
  - Specifies the maximum number of alert records that are kept in storage. The
NPDA.ALCACHE

Valid range for maxnumber is 10–9999. The default is 10. The greater the value of maxnumber, the more efficient your hardware monitor performance is for operators viewing the Alerts-Dynamic panel.

NONE

Specifies that no storage is allocated for alert records.

WRAPCNT

Specifies that the maximum number of alert records kept in storage is the same as the current alert wrap count. WRAPCNT is the most efficient setting, because whenever the hardware monitor updates an operator’s Alerts-Dynamic panel, the alert can be retrieved from the alert cache.

Usage Notes:

The alert cache is set to the smaller of one of the following:

- The size specified by the NPDA.ALCACHE statement
- The current wrap count

The size of your alert cache is equal to the number of alert records in storage.

The alert cache serves two purposes:

- Provides improved performance time for operators viewing the Alerts-Dynamic panel.
  - By keeping the alerts in the alert cache, you can decrease the amount of VSAM input/output (I/O) and processor time (decreasing the processor utilization percentage for the BNJDSERV task) when the Alerts-Dynamic panel is updated. This performance improves when:
    - More operators view the Alerts-Dynamic panel; each operator, who views the Alerts-Dynamic panel, obtains improved performance time. The more operators who view this panel, the greater the performance improvement.
    - Your system receives more alerts. The more alerts your system receives, the more performance improves.
  - Alerts, received during a time when the NetView program has no SNA-MDS/LU 6.2 alert focal point, are marked as held in the alert cache. Later when the focal point is reacquired, the alerts that are being held are forwarded to the focal point.

If your operators seldom view the Alerts-Dynamic panel, you do not obtain a significant performance improvement, regardless of the ALCACHE setting (maxnumber or WRAPCNT).

Table 10. Examples of the NPDA.ALCACHE Statement

<table>
<thead>
<tr>
<th>ALCACHE Statement</th>
<th>Alert Wrap Count</th>
<th>Alert Cache Count</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPDA.ALCACHE = 500</td>
<td>100</td>
<td>100</td>
<td>Because the alert wrap count is smaller than the value specified with ALCACHE, the alert cache size is the same as the alert wrap count.</td>
</tr>
</tbody>
</table>
Table 10. Examples of the NPDA.ALCACHE Statement (continued)

<table>
<thead>
<tr>
<th>ALCACHE Statement</th>
<th>Alert Wrap Count</th>
<th>Alert Cache Count</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPDA.ALCACHE = 500</td>
<td>500</td>
<td>500</td>
<td>Because the alert wrap count is the same size as the value specified with ALCACHE, the alert cache size is 500.</td>
</tr>
<tr>
<td>NPDA.ALCACHE = 100</td>
<td>999</td>
<td>100</td>
<td>Because the value specified with ALCACHE is smaller than the alert wrap count, the alert cache size is 100.</td>
</tr>
<tr>
<td>NPDA.ALCACHE = NONE</td>
<td>100</td>
<td>0</td>
<td>The alert cache is not used to store alerts.</td>
</tr>
<tr>
<td>NPDA.ALCACHE = WRAPCNT</td>
<td>500</td>
<td>500</td>
<td>The alert cache is set to the size of the alert wrap count.</td>
</tr>
</tbody>
</table>

To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: NPDA.ALERTFWD

**NPDA.ALERTFWD**

The NPDA.ALERTFWD statement defines the alert forwarding protocol that this NetView host will use. This statement is coded in CNMSTYLE.

For more information about alert forwarding, refer to the [Tivoli NetView for z/OS Automation Guide](#).

The syntax for the NPDA.ALERTFWD statement is:

NPDA.ALERTFWD

```plaintext
NPDA.ALERTFWD= SNA-MDS-LOGONLY
  SNA-MDS-AUTHRCV
  SNA-MDS-SUPPRESS
  NV-UNIQ
```

Where:

**SNA-MDS**

Any one of these options (LOGONLY, AUTHRCV, SUPPRESS) designates that NetView is to forward alerts over the LU 6.2 transport using the architected SNA-MDS/LU6.2 alert forwarding protocol (refer to the [Systems Network Architecture library](#)). Using SNA-MDS is recommended, except when the focal point is prior to NetView V3. SNA-MDS/LU 6.2 alert forwarding supports NetView and non-NetView entry points and focal points. The alert-forwarding function also supports intermediate node focal points.

When SNA-MDS is specified, the DEFFOCPT statements for category ALERT in the DSI6DST task initialization member DSI6INIT are accepted and
processed. DEFFOCPT statements that are for category ALERT in the DSICRTR task initialization member DSICRTTD are ignored, rejected, and message BNH096I is issued.

The various options supported by SNA-MDS indicate whether or not to issue a message when the following situations occur:

- The ALERT focal point is a non-NetView entry point.
- The entry point NetView forwards an unarchitected alert to the non-NetView focal point.
- The non-NetView focal point responded by sending an MDS error message or an application error message to the NetView entry point.

The following three SNA-MDS options specify how the MDS error message or application error message is to be processed:

**SNA-MDS-LOGONLY**
- Causes message BNH094I or BNH095I to be placed in the NetView log.
- LOGONLY is the recommended option.

**SNA-MDS-AUTHRCV**
- Issues message BNH094I or BNH095I to the NetView authorized receiver task.

**SNA-MDS-SUPPRESS**
- Does not issue BNH094I or BNH095I error messages.

**NV-UNIQ**
- Designates that NetView is to forward alerts, as it does in releases prior to V3, using the NetView-to-NetView NV-UNIQ/LUC alert forwarding protocol over the LUC transport. This forwarding protocol does not support intermediate node focal points, nor does it support non-NetView focal points such as an AS/400.

When NV-UNIQ is specified, the DEFFOCPT statements for category ALERT in the DSICRTR task initialization member DSICRTTD are accepted and processed. Any DEFFOCPT statements for category ALERT in the DSI6DST task initialization member DSI6INIT are ignored and rejected with message BNH096I.

**Usage Notes:**

Only code the NPDA.ALERTFWD statement if you are defining a NetView host as an intermediate node or as an entry-point NetView host that forwards alerts to an alert focal point.

You cannot use different alert forwarding protocols at the same time. Examine the alert focal point hierarchy of your network to determine which setting to use, SNA-MDS or NV-UNIQ. Refer to the [Tivoli NetView for z/OS Automation Guide](https://www.ibm.com) for information on determining which alert-forwarding protocol to use.

The ALERTFWD setting affects the following commands for only the ALERT category:

- **FOCALPT ACQUIRE FPCAT=ALERT**
  - When this command is entered at the entry point, NetView acquires an SNA-MDS focal point or an NV-UNIQ focal point.
- **FOCALPT CHANGE FPCAT=ALERT** (at entry point only)
When this command is entered at the focal point, NetView acquires an SNA-MDS entry point or an NV-UNIQ entry point.

- **CHANGEFP for category ALERT (at entry point only)**
  When this command is entered at the focal point, NetView acquires an NV-UNIQ entry point.

- **FOCALPT DROP FPCAT=ALERT**
  When this command is entered at the entry point, NetView drops an SNA-MDS focal point or an NV-UNIQ focal point.

- **FOCALPT QUERY FPCAT=ALERT**
  When this command is entered at the entry point, NetView displays either the SNA-MDS focal point or the NV-UNIQ focal point.

To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

**Related Statements:** DEFFOCPT, NPDA.ALCACHE, NPDA.ALRTINFP

---

**NPDA.ALERTLOG**

The NPDA.ALERTLOG statement controls the way the hardware monitor writes records to the hardware monitor VSAM database. This statement allows a choice between VSAM performance and CPU processor time by changing the way the hardware monitor writes records to the database. It is processed only when the hardware monitor starts (for example, START TASK=BNJDSERV) with a newly allocated database. After that, the NPDA.ALERTLOG statement is ignored, even if the database is empty. Also, if you SWITCH to a newly allocated database, the NPDA.ALERTLOG setting, which was present when the BNJDSERV task was started, is the setting used with the new database. The NPDA.ALERTLOG statement is not read again on a SWITCH. The NPDA.ALERTLOG statement is coded in CNMSTYLE.

The syntax for the NPDA.ALERTLOG statement is:

```
NPDA.ALERTLOG
```

Where:

**RANDRANG**

Specifies that alert records and non-alert records (event records and correlation records, for example) are logged in a single key range. This is the most efficient setting for recording records to the VSAM database, and results in the fewest VSAM control interval and control area splits. For most customers, RANDRANG is the recommended setting. However, depending on the contents of the hardware monitor database, hardware monitor initialization might take several minutes to complete because all records (both alert and non-alert) in the database are read. Hardware monitor initialization in this context includes when the BNJDSERV task is started (for example, START TASK=BNJDSERV) and when the active database is changed (for example, SWITCH BNJDSERV).
NPDA.ALERTLOG

RSTDRANG
Specifies that alert records are logged in one key range in the database, and non-alert records are logged in another key range in the database. This setting is less efficient for VSAM; more VSAM control interval splits and VSAM control area splits can occur when alert records are recorded to the database. However, depending on the contents of the hardware monitor database, hardware monitor initialization can sometimes proceed more quickly with RSTDRANG than with RANDRANG. During initialization, all alert records in the database are read. When alerts are logged in a restricted key range, the records can sometimes be read much faster. Use the RSTDRANG setting if you find that hardware monitor initialization is taking too long to complete with the default RANDRANG setting.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

NPDA.ALRTINFP

The NPDA.ALRTINFP statement specifies whether SNA-MDS/LU 6.2 forwarded alert records and whether event/statistic records for non-NetView entry points should be recorded to the hardware monitor database at the NetView host before they are forwarded to the alert focal point.

The NetView ALERT-NETOP function for the hardware monitor supports intermediate nodes. For example, an entry point ALERT-NETOP can forward an alert to an intermediate node ALERT-NETOP, and the intermediate node ALERT-NETOP can forward the alert again.

The NPDA.ALRTINFP statement is valid only when all of the following conditions are true; otherwise, it is not applicable, and its setting is ignored:

1. The LU 6.2 alert forwarding protocol is being used to forward alerts. The NPDA.ALERTFWD statement with the SNA-MDS/LU 6.2 option must be defined in CNMSTYLE.
2. Alerts must be received from another node (either NetView or non-NetView) over the LU 6.2 alert forwarding protocol. The NPDA.ALRTINFP setting is not applicable for local alerts or for NV-UNIQ/LUC forwarded alerts.
3. This NetView host is an intermediate node focal point for the category ALERT that currently has an active alert focal point defined with the LU 6.2 alert forwarding protocol.

Note: If the NetView host alert focal point is unavailable, this NPDA.ALRTINFP setting is ignored until an ALERT focal point is reacquired. LU 6.2 alerts that are forwarded and received during this time are recorded to the hardware monitor database, just as they are at a non-intermediate node NetView focal point.

The NPDA.ALRTINFP statement is coded in CNMSTYLE. For more information on architected alert forwarding, refer to the Tivoli NetView for z/OS Automation Guide.

The syntax for the NPDA.ALRTINFP statement is:
NPDA.ALRTINFP

NPDA.ALRTINFP.RECORD = YES

Where:

YES

Specifies that the hardware monitor records all forwarded alerts it receives (using LU 6.2) from entry points to the hardware monitor database. YES is the recommended setting.

NO

Specifies that the hardware monitor not record the alerts it receives that were forwarded (using LU 6.2) from entry points to the hardware monitor database.

For intermediate focal point nodes, using NO is beneficial because the intermediate node’s hardware monitor database does not record unnecessary data that is processed by the ultimate focal point. The alert passes through this node without being recorded. Also, processor time is not consumed while logging unnecessary records. Alerts are still submitted to automation, even though they are not recorded to the database.

Usage Notes:

- Code the NPDA.ALRTINFP statement only if you are defining a NetView host as an intermediate node.

  If an intermediate node’s alert focal point is unavailable, all alerts processed during the time that the focal point is unavailable are processed as if the intermediate node is the ultimate focal point. The alerts are recorded to the hardware monitor database, and the NPDA.ALRTINFP statement is ignored.

- NPDA.ALRTINFP.RECORD = Yes is the initial setting in CNMSTYLE.

- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: NPDA.ALT_ALERTFWD, DEFFOCPT

NPDA.ALT_ALERT

The NPDA.ALT_ALERT statement enables you to define the data that is displayed on the Alerts-Dynamic (NPDA-30A), Alerts-Static (NPDA-30B), and Alerts-History (NPDA-31A) hardware monitor panels.

The NPDA.ALT_ALERT statement is coded in CNMSTYLE.

The syntax for the NPDA.ALT_ALERT statement is:

NPDA.ALT_ALERT

Where:
NPDA.AUTORATE

The NPDA.AUTORATE statement enables management services units (MSUs) that are blocked as a result of a filter from the NPDA.RATE statement to be automated. See "NPDA.RATE" on page 133 for more information.

The NPDA.AUTORATE statement is coded in CNMSTYLE. Changes to CNMSTYLE containing the NPDA.AUTORATE statement do not take effect until you execute the RESTYLE command with the NPDA option. If you change CNMSTYLE and the change results in additional extents being used, stop and restart the NetView program to implement the change.

Notes:

1. If you do not code an NPDA.AUTORATE statement, MSUs blocked by a filter set by the RATE function are not automated.

2. XITCI return code 252 indicates that an MSU should be logged to the system management facilities (SMF) only. Because these MSUs are not recorded as events in the hardware monitor database, the NPDA.RATE function might not set a filter for them and they might not be affected by the NPDA.AUTORATE statement.

The syntax for the NPDA.AUTORATE statement is:

NPDA.AUTORATE

Where:

1. Indicates that MSUs blocked as a result of the RATE function are not sent to automation. This is the initial setting in CNMSTYLE.

2. Indicates that MSUs blocked as a result of the RATE function are sent to automation.

3. Indicates that MSUs blocked as a result of the RATE function are not sent to automation. However, MSUs with an XITCI return code of 252 or 253 are automatically sent to automation.

Related Statements: NPDA.RATE
NPDA.DSRBO

The NPDA.DSRBO statement specifies the number of DSRBOs to be associated with the BNJDSERV task. This statement is coded in CNMSTYLE.

The syntax for the NPDA.DSRBO statement is:

```
NPDA.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 5.

Usage Notes:
- If you change the `number`, update the STRNO keyword on the BLDVRP macro used to create the LSR pool in CNMSJM01.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

NPDA.DSRBU

The NPDA.DSRBU statement specifies the number of data services request blocks (DSRBs) to pre-allocate for processing unsolicited problem determination request units (RUs). This statement is coded in CNMSTYLE.

The syntax for the NPDA.DSRBU statement is:

```
NPDA.DSRBU=number
```

Where:

- `number` is a decimal number from 0–999. The default value is 5.

To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

NPDA.ERR_RATE

The NPDA.ERR_RATE statement enables you to regulate the generation of messages for alerts that are not valid.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.ERR_RATE statement is:
NPDA.ERR_RATE

\[
\text{NPDA.ERR_RATE} = \begin{bmatrix}
10 & \text{mmm} \\
50 & \text{iii}
\end{bmatrix}
\]

Where:

10 | mmm
Identifies the maximum number of messages for alerts that can be generated in a one-minute window before further messages are suppressed. Suppression continues until the number of messages per minute drops below this value. The valid range for this value is 000–999. A value of 000 results in no suppression. The default value is 10.

50 | iii
Indicates the number of alerts to be suppressed in a one-minute window before message BNJ040I is issued. The valid range for this value is 001–999. The default value is 50.

Usage Notes:
- You must specify both the maximum number of messages for alerts and the number of alerts to be suppressed on the NPDA.ERR_RATE statement.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

NPDA.IHTHRESH

The NPDA.IHTHRESH statement enables you to set thresholds for the number of impulse hits (amplitude hits) in leased lines connected to IBM LPDA-2 modems. If a modem experiences line problems and is put into backup speed, the NetView program uses the impulse hits and line quality thresholds to determine when the line can be returned to full speed. When the impulse hits and line quality thresholds are lower than the thresholds specified, the NetView program issues message BNJ018I. When this message is issued, you can issue the MDMCNTL command manually or through automation to return the line to full speed.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.IHTHRESH statement is:

\[
\text{NPDA.IHTHRESH} = \begin{bmatrix}
nnn
\end{bmatrix}
\]

Where:

nnn
Indicates the threshold for the number of impulse hits. The threshold can be a whole number in the range of 0–255. The default is zero (0).

Usage Notes:
- The suggested threshold is 12.
If the NPDA.IHTHRESH statement is commented out in CNMSTYLE, the threshold is set to 0. When the threshold is 0, the NetView program does not issue message BNJ018I.

Message BNJ018I is issued when a statistical record from NCP indicates that the line quality and impulse hits are lower than the specified thresholds.

To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: NPDA.LQTHRESH

NPDA.LQTHRESH

The NPDA.LQTHRESH statement enables you to set thresholds for the quality of leased lines connected to IBM LPDA-2 modems. If a modem experiences line problems and is put into backup speed, the NetView program uses the impulse hits and line quality thresholds to determine when the line can be returned to full speed. When the impulse hits or line quality are below the thresholds specified, the NetView program issues message BNJ018I. When this message is issued, you can issue the MDMCNTL command manually or through automation to return the line to full speed.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.LQTHRESH statement is:

```
NPDA.LQTHRESH

   NPDA.LQTHRESH = nnn

Where:

   nnn

   Indicates the threshold for the quality of the leased line. The threshold can be a whole number in the range of 0-15. The default is 5.

Usage Notes:

   • The NetView program issues message BNJ018I when a statistical record from NCP indicates that the line quality and impulse hits are lower than the specified thresholds.
   • To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: NPDA.IHTHRESH

NPDA.MACRF

The NPDA.MACRF statement specifies the local shared resource (LSR) options.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.MACRF statement is:
**NPDA.MACRF**

<table>
<thead>
<tr>
<th>NPDA.MACRF</th>
</tr>
</thead>
</table>
| DFR
| LSR
| NSR |

**Where:**

**DFR**

Extends LSR to defer writing of records. The deferred write (DFR) option defers the writing of a record until the NetView program forces it out because buffer space is needed for a read. This further reduces I/O by minimizing writes.

**Attention:** If DFR is coded, do not use the MVS operator CANCEL command or the MVS STOP FORCE command to end the NetView task. Issuing these commands can damage the NetView databases by preventing the final writing of records for which output was deferred. To end the NetView task, use the NetView CLOSE or CLOSE IMMED commands.

**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 default.

**NSR**

Indicates that the data set does not use shared resources. This option is not recommended.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

---

**NPDA.MDSIND**

The NPDA.MDSIND statement specifies whether the at sign (@) should be displayed on alert panels to indicate that an alert was forwarded by the SNA-MDS method. This statement is coded in CNMSTYLE.

The syntax for the NPDA.MDSIND statement is:

<table>
<thead>
<tr>
<th>NPDA.MDSIND = YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
</tr>
</tbody>
</table>

**Where:**

**YES**

Specifies that the at sign is displayed.

**NO**

Specifies that the at sign is not displayed.

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.
**NPDA.PDDNM**

The NPDA.PDDNM statement specifies the hardware monitor primary data set. This statement is coded in CNMSTYLE.

The syntax for the NPDA.PDDNM statement is:

```
NPDA.PDDNM=name
```

Where:

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

Usage Notes:
- If necessary, specify the VSAM password in CNMSTPWD.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

**NPDA.PNA**

The NPDA.PNA statement specifies whether you use PNA programs in your network. PNA is an application that runs on a workstation and acts as a gateway through which NetView commands can be issued to physical units downstream. PNA enables communication between SNA and non-SNA communication systems. This statement is coded in CNMSTYLE.

The syntax for the NPDA.PNA statement is:

```
NPDA.PNA=YES
```

Where:

YES
Specifies that you use PNA support.

NO
Specifies that you do not use PNA support.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

**NPDA.PRELOAD**

The NPDA.PRELOAD data is used if your network or system contains devices or applications that send Basic Encoding Rules (BER) data to the NetView program.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.PRELOAD statement is:
Where:

YES
Specifies that BER data is sent to the NetView program.

NO
Specifies that BER data is not sent to the NetView program. No is the default.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

NPDA.R (Ratio)

The NPDA.R (Ratio) statement enables you to assign initial or default ratio values when the first error record for a particular resource is received. These statements do not alter the wrap and ratio counts for resources already on the hardware monitor database.

To change the wrap and ratio values for existing resources, use the explicit SWRAP and SRATIO command. Refer to the NetView online help for more information about these commands. The altered values remain in effect until one of the following occurs:

- The hardware alerts database is reinitialized.
- All hardware alert records for the particular resource and all attached resources are purged.
- The values are changed by another SWRAP or SRATIO command.

This statement is coded in CNMSTYLE. To implement changes to CNMYSTYLE containing the NPDA.R statement, execute the RESTYLE command with the NPDA option. If changing CNMYSTYLE results in additional extents being used, stop and restart the NetView program.

The syntax for the R statement is:

```
NPDA.R (RATIO)
```

Where:

- **suffix**
  A required identifier that makes an NPDA.R (RATIO) statement unique. Use a different suffix for each R statement coded.

- **rtype**
  Indicates the resource type. The resource type must conform to the following:
  - For a level 2 resource hierarchy, the only valid resource types are CBUS, FRLY, and LAN.
For a level 3 resource hierarchy, all resource types are valid.
For a level 4 resource hierarchy, all resource types are valid except LINE.
For a level 5 resource hierarchy, all resource types are valid.

You can specify the following resource types:
- **LINE** Is a line
- **CTRL** Is a cluster controller on the remote end of a line
- **LCTL** Is a cluster controller attached to the CPU
- **LDEV** Is a device attached to the remote cluster controller
- **LINK_ATT** Is a link-attached resource
- **CHAN_ATT** Is a channel-attached resource

The `rname` indicates the resource name. This name is a single name that corresponds to the resource type. If several resources have the same type and name, they are all affected. The `rname` value is required. If `rtype` is LINK_ATT or CHAN_ATT, `rname` must be an asterisk (*).

The `v`v`v` indicates the initial error-to-traffic (E/T) ratio specified for `rtype`. The maximum E/T ratio value that you can enter is 250 (interpreted by the NetView program as 25.0%). Unspecified resources are assigned appropriate defined default E/T ratios. If the default E/T ratios are not defined, the NetView program uses 3.0% for all unspecified link-attached resources and 1.0% for all unspecified channel-attached communication resources.

**Related Statements:** NPDA.W

---

**NPDA.RATE**

The NPDA.RATE statement enables you to set the maximum rate at which events can be logged to the hardware monitor database. This function stops database logging of repetitive events from a resource. The RATE function compares the time between an event being wrapped off the database and the new event record being recorded on the database. A rate filter is set to block the recording of events from the resource if the difference is less than the time specified on the RATE statement. The RATE function takes no action if the time difference of the new event record is previous to the record that is about to be overwritten. This prevents the RATE function from setting an excessive number of filters due to a damaged database (caused by setting the wrong date when initializing the system).

This statement is coded in CNMSTYLE.

The syntax for the NPDA.RATE statement is:

```
NPDA.RATE
```

Where:

* `mm:ss`
  Identifies the number of minutes and seconds that must elapse before an event
NPDA.RATE

can wrap off the database after it is recorded. If the event record wraps off the
database in less time than $mm:ss$, a RATE filter is set for the resource name.

Usage Notes:

- If you do not use the NPDA.RATE statement, the rate value is set to zero (0). A
  value of 0 turns off the RATE function.
- Message BNJ045I is issued when the rate filter is set. When the cause of the
  excessive events is determined, delete the filter entry.
- If you set your wrap count at 25, you receive the recommended wrap rate count
  of 1 per second. Low wrap counts can cause the filter to be set and the message
  to be issued erroneously.
- The RATE function is not performed on events with a wrap count of 0 or 1.
- An event that is blocked by a recording filter set by the RATE function is not
  sent to automation. If you want these events automated, use an AUTORATE
  statement in CNMSTYLE. See "NPDA.AUTORATE" on page 126 for more
  information about whether events that are blocked by a recording filter set by
  the rate function are automated.
- To implement definition changes while the NetView program is active, change
  the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: NPDA.AUTORATE, NPDA.W

NPDA.REPORTS

The NPDA.REPORTS statement specifies whether report logging starts at hardware
monitor initialization. These report records can be written to the system
management facilities (SMF) data set or to a data set defined through a NetView
program exit. Generate management reports from this file with the Service Level
Reporter (SLR) licensed program.

Change the report-logging setting any time after initialization by issuing the
REPORTS command.

Data from input records that results in multiple records written to the database is
recorded in a single report record. The external log record formats are in the Tivoli
NetView for z/OS Application Programmer’s Guide.

This statement is coded in CNMSTYLE.

The syntax for the NPDA.REPORTS statement is:

NPDA.REPORTS

```
OFF
-ON-
-XLO
```

Where:

- OFF
  Means that record logging does not start automatically at initialization. OFF is
  the default.
ON
Means that record logging starts automatically when the hardware monitor is initialized. Records are logged for each hardware monitor input record for which both of the following are true:
• Report logging is supported.
• The event or statistical recording (ESREC) filter was passed or external logging only was specified through either the XITCI installation exit or automation processing.

XLO
Means that record logging starts automatically when the hardware monitor is initialized. Records are logged for each hardware monitor input record for which both of the following are true:
• Report logging is supported.
• External logging only was specified through either the XITCI installation exit or automation processing.

Usage Notes:
• To run SLR against SMF logs that the NetView program has written in a double-byte character set (DBCS), obtain a release of SLR that supports DBCS.
• To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

Related Statements: DSTINIT, TASK

NPDA.SDDNM
The NPDA.SDDNM statement specifies hardware monitor secondary data set. This statement is coded in CNMSTYLE.

The syntax for the NPDA.SDDNM statement is:

```
NPDA.SDDNM=name
```

Where:

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

Usage Notes:
• If necessary, specify the VSAM password in CNMSTPWD.
• To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.
The NPDA.TECROUTE statement specifies the name of the PPI receiver associated with the event/automation service. This statement is coded in CNMSTYLE.

The syntax for the NPDA.TECROUTE statement is:

```
NPDA.TECROUTE = name
```

Where:

- **name**
  - The PPI receiver name. The default value is IHSATEC.

Usage Notes:
- If necessary, specify the VSAM password in CNMSTPWD.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

### NPDA.W (Wrap)

The NPDA.W (Wrap) statement enables you to assign initial wrap values when the first error record for a particular resource is received. These statements do not alter the wrap and ratio counts for resources existing on the hardware monitor alerts database.

You can specify initial wrap count values (**nnn**) shown on the W statements for event (EV), statistical (ST), alert (AL), MSU correlation (MSU_CORR), resource correlation (RES_CORR), and GMFALERT data types. You can specify a value for only one data type on a W statement; therefore, to specify values for multiple data types, use multiple W statements. The values for wrap counts are set in the CNMSTYLE file. These values are called from CNMSTYLE by the BNJMBDST sample file when the NPDA.W statement is run.

Event and statistical wrap count statements must indicate the resource type (**rtype**). The alert (AL) statement in CNMSTYLE sets wrap counts for alerts on the hardware monitor and NPDA database.

The NPDA.W and NPDA.R statements assign initial wrap and ratio values when the first error record for a particular resource is received. Once the initial wrap values are set, the NPDA.W statement is not read again.

To change the wrap and ratio values for existing resources, use the explicit SWRAP and SRATIO commands (or reinitialize the hardware monitor database). Refer to the NetView online help for more information about these commands. The altered values remain in effect until one of the following occurs:
- The hardware monitor database is reinitialized.
- All hardware monitor records for the particular resource and all attached resources are purged.
- The values are changed by another SWRAP or SRATIO command.

Note that the MSU, resource correlation, and GMFALERT wrap counts cannot be changed by the SWRAP command.
This statement is coded in CNMSTYLE.

The syntax for the NPDA.W statement is:

```
NPDA.W (Wrap)
```

```
NPDA.W.suffix = AL
               EV
               ST
             MSU_CORR
            RES_CORR
          GMFALERT
```

Where:

`suffix`
A required identifier that makes a NPDA.W (wrap) statement unique. Use a different suffix for each wrap statement coded.

`AL`
Indicates the alert data type.

```
100|nnnn
```
Selects the initial wrap count value, in the range of 0–9999, for AL. The default value is 100.

`EV`
Indicates the event data type.

```
025|nnn
```
Selects the initial wrap count value, in the range of 0–450, for EV. The default value is 025.

`rtype`
You can access help that lists the resource types by entering HELP from any hardware monitor panel. From the Help menu, enter 3 to access the Common Format panel. A list of resource types begins at panel 4.

`ST`
Indicates the statistical data type.

```
025|nnn
```
Selects the initial wrap count value, in the range of 0–450, for ST. The default value is 025.

`rtype`
You can access any help that lists the resource types by entering HELP from a hardware monitor panel. From the Help menu, enter 3 to access the Common Format panel. A list of resource types begins at panel 4. To access this panel more quickly, you can enter ? TYPE, but this does not work from panels that have the term TYPE as part of a heading of the particular panel.

`MSU_CORR`
Indicates the MSU correlation data type.
Selects the initial wrap count value, in the range of 0–450, for MSU correlation records. The hardware monitor creates these records when the input record received by the hardware monitor contains one or more MSU correlation X'47' subvectors or one or more incident identification X'4A' subvectors. These records correlate event records to other event records. The hardware monitor uses MSU records when building the Correlated Events panel (NPDA-41B). (Refer to the SNA library for more information regarding these subvectors.)

The wrap count applies to each unique correlator. For example, if the wrap count is 25 and the hardware monitor receives 20 input records that contain correlator A and 25 input records that contain correlator B, the hardware monitor creates 20 MSU correlation records for correlator A and an additional 25 MSU correlation records for correlator B. If a 26th input record contains correlator B, the wrap count of 25 for correlator B takes effect, and the oldest MSU correlation record for B is erased.

The default value is 025.

RES_CORR
Indicates the resource correlation data type.

Selects the initial wrap count value, in the range of 0–450, for resource correlation records. For each input record that it receives, the hardware monitor determines the name of the failing resource and creates a resource correlation record against the failing resource name. Also, if the input record contains a hierarchy/resource list X'05' subvector with associated resources hierarchy/resource list X'11' subfield, then the hardware monitor creates resource correlation records against each resource name present in the subfield X'11' resource list. These records correlate event records to other event records. The hardware monitor uses these records when building the Correlated Events by Resource panel (NPDA-41B).

This wrap count applies to each unique resource name. For example, if the wrap count is 25 and the hardware monitor receives 20 input records that contain failing resource name A and 25 input records that contain failing resource name B, the hardware monitor creates 20 resource correlation records for name A and an additional 25 records for B. If a 26th input record is received that contains failing resource name B, the wrap count of 25 for name B takes effect, and the oldest resource correlation record for B is erased.

The default value is 025.

GMFALERT
Indicates the number of records stored in the NPDA database for each RODM monitored by NetView management console.

Selects the initial wrap count value, in the range of 0–256, for each resource monitored by NetView management console.

The default value is 025.

Usage Notes:
- For resource types LAN and RING, the default wrap count is 100.
NPDA.W (Wrap)

- Specifying a wrap count for rtype only affects events for which the lowest (rightmost) level of the alert hierarchy matches the value of rtype.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NPDA command.

**Related Statements:** TASK, NPDA.RATE

---

**NQNSUP**

The NQNSUP statement controls whether the NetView Management Services transport running on this NetView program will use network qualified names on all requests to send and receive Management Services data. You code the NQNSUP statement in CNMSTYLE.

The syntax for the NQNSUP statement is:

```
NQNSUP
```

Where:

- **NO**
  - Specifies that an LU name is sufficient on LU 6.2 APIs. NO is the default.
- **YES**
  - Specifies that the netid can be provided on LU 6.2 send requests. If no netid is specified, the local network name is used. This support allows duplicate LU names in other networks communicating with this NetView program.

**Usage Note:** If you change the value of NQNSUP after the NetView program is initialized, the NetView program must be recycled for changes to take effect.

---

**NRM.CMODE**

The NRM.CMODE statement specifies the mode to communicate between NetView hosts for the NetView Resource Manager function.

This statement is coded in CNMSTYLE.

The syntax for the NRM.CMODE statement is:

```
NRM.CMODE
```

Where:

- **number**
  - A numerical value, indicating that multiple NRM.CMODE values can be specified.
**NRM.CMODE**

- **IP** Indicates to use IP communication mode.
- **SNA** Indicates to use SNA communication mode.

**Usage Notes:**
- If you want to use the same communication mode for all host destinations, then one value of SNA or IP can be specified. If both SNA and IP communication modes are used, then you must specify an NRM.CMODE value for each NRM.HOSTDEST value.
- NRM.CMODE is ignored if NRM.HOSTDEST is not specified.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

**NRM.CONNINT**

The NRM.CONNINT statement specifies the interval between connection retries when establishing communication between the manager and agent hosts.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.CONNINT statement is:

```
NRM.CONNINT

NRM.CONNINT=seconds
```

*Where:*

- **seconds** Specifies the number of seconds in the range 0–1440. If this value is 0, the retries occur one after the other with no time delay. The default is 180.

**Usage Notes:**
- NRM.CONNINT is used in conjunction with NRM.CONNRETRY.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

**NRM.CONNRETRY**

The NRM.CONNRETRY statement specifies the number of connection retries to be attempted when communication between the manager and agent fails.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.CONNRETRY statement is:

```
NRM.CONNRETRY

NRM.CONNRETRY=retries
```

140 Administration Reference
Where:

retries
    Specifies a value is in the range 0–10. The default is 3.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.HBRETRY

The NRM.HBRETRY statement specifies how often a manager tests connectivity to its agents.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.HBRETRY statement is:

NRM.HBRETRY

Where:

seconds
    Specifies the number of seconds in the range 30–600. The default is 30.

Usage Notes:
• NRM.HBRETRY is valid only with NRM.TYPE=MGR.
• When NRM determines that connectivity is lost, the status for all resources for that domain is UNKNOWN.
• To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.HOSTDEST

The NRM.HOSTDEST statement specifies the manager host that shows status for this agent host. HOSTDEST can be specified for a manager or agent host.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.HOSTDEST statement is:

NRM.HOSTDEST

Where:

number
    A numerical value, indicating that one or more host destinations can be specified.
**NRM.HOSTDEST**

*destination*

If the communication mode (see NRM.CMODE) between this agent and the manager host is SNA, the value for destination is the NetView domain of the manager. If the communication mode between this agent and the manager host is IP, the value for destination is a dotted decimal IP address or a host name.

**Usage Notes:**
- NRM.HOSTDEST is required if NRM.TYPE=AGENT is specified.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

**NRM.PORT**

The NRM.PORT statement specifies the port number for the DSIRTTR task at each NRM.HOSTDEST.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.PORT statement is:

```
NRM.PORT

Where:

number
   A numerical value, indicating that one or more host destinations can be specified.

port_number
   Specify a value in the range 1–65535. The default is 4021.

Usage Notes:
- If only one port number is specified, that port number is used for all IP host destinations.
- If both SNA and IP communication modes are used, specify an NRM.PORT value for each NRM.HOSTDEST value that uses IP communications.
- NRM.PORT is only valid when NRM.CMODE=IP. When you specify NRM.HOSTDEST with NRM.CMODE=SNA, leave the NRM.PORT statement commented out.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.
```

**NRM.RESET**

The NRM.RESET statement specifies if you want all NetView Resource Manager objects set to an initial status at NRM initialization.

This statement is coded in CNMSTYLE.

The syntax for the NRM.RESET statement is:
NRM.RESET

Where:

NO
Indicates that no initial status will be set during NRM initialization. This is the default.

Note: NRM objects will be updated to their correct status when an agent communicates status to the manager.

YES
Indicates that the status of objects should be set to the status specified by NRM.STATUS.RESET under the DISPLAY STATUS section in CNMSTYLE.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.SAMPLERATE

The NRM.SAMPLERATE statement specifies the frequency in requesting complete data for this NetView Resource Manager host.

This statement is coded in CNMSTYLE.

The syntax for the NRM.SAMPLERATE statement is:

NRM.SAMPLERATE

Where:

minutes
Specifies the number of minutes and is in the range 0–1440. The default is 0, no sampling.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.STATUS

The NRM.STATUS statement specifies the display status values that should be used for NetView Resource Manager objects.

This statement is coded in CNMSTYLE.

The syntax for the NRM.STATUS statement is:
NRM.STATUS

Where:

state
Indicates the NRM object state. For more information, refer to CNMSTYLE.

status
Indicates the display status value. For more information, refer to CNMSTYLE.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.TYPE

The NRM.TYPE statement specifies whether this host is a manager or an agent host.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.

The syntax for the NRM.TYPE statement is:

NRM.TYPE

Where:

AGENT
Indicates that this host forwards local host information to one or more manager hosts.

MGR
Indicates that this host stores information in RODM for viewing at the NetView management console. This is the default.

Usage Notes:
• If MGR is specified, a RODMname value must also be specified.
• By default, a manager is an agent for itself.
• To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

NRM.XCLDOM

The NRM.XCLDOM statement specifies which domains the manager host should not monitor.

This statement is coded in CNMSTYLE and is used for the NetView Resource Manager function.
The syntax for the NRM.XCLDOM statement is:

```
NRM.XCLDOM
```

Where:

- **domainid**
  - Specifies a domain name with a value 1–5 characters in length. Wildcards can be specified for the domain names.

Usage Notes:
- The parentheses are not required if only one value is specified. Multiple values must be enclosed in parentheses and separated by either blanks or commas.
- NRM.XCLDOM is only applicable when NRM.TYPE=MGR.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

---

**NRM.XCLSRC**

The NRM.XCLSRC statement specifies which set of exclusion lists, if any, that you want to use with the NetView Resource Manager.

This statement is coded in CNMSTYLE.

The syntax for the NRM.XCLSRC statement is:

```
NRM.XCLSRC
```

Where:

- **CTL**
  - Indicates to use the exclusion lists that were already in effect. This enables you to retain exclusion lists that you have defined dynamically across NetView Resource Manager calls.

- **VARS**
  - Indicates to use the exclusion lists as defined in CNMSTYLE. Specifying VARS overrides any exclusion lists that were previously set. This is the default.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.
The NRM.XCLTASKN statement specifies which tasks should not be monitored by the NetView Resource Manager.

This statement is coded in CNMSTYLE.

The syntax for the NRM.XCLTASKN statement is:

```
NRM.XCLTASKN
```

Where:

`task_name`

Specifies a task name 1–8 characters in length. Wildcards can be specified for the task names.

Usage Notes:

- The parentheses are not required if only one task is specified. Multiple values must be enclosed in parentheses and separated by either blanks or commas.
- NRM.XCLTASKN is a function of the NRM agent and is only applicable on the local NetView host.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE NRM command.

The NRM.XCLTASKT statement specifies which types of tasks the NetView Resource Manager should not monitor. For example, specifying NRM.XCLTASKT=OST excludes all operator station tasks from monitoring by NRM.

This statement is coded in CNMSTYLE.

The syntax for the NRM.XCLTASKT statement is:

```
NRM.XCLTASKT
```

Where:

`task_type`

Valid task types are as follows:

- DST
- HCT
- MNT
O MONIT

The O MONIT statement enables the status monitor to reactivate inactive nodes other than major nodes, applications, cross-domain resources, and resources past the local NCP. You can exclude nodes from automatic reactivation as follows:

- By flagging them with the NOMONIT operand on the STATOPT statement
- By using the MONOFF command or the MONIT STOP command to add them to the node reactivation exclusion list

You code the O MONIT option control statement in DSICNM. Changes to DSICNM do not take effect until you stop and restart the VMT task. If changing the member results in additional extents being used, you need to stop and restart the NetView program before the changes take effect.

The syntax for the O MONIT statement is:

```
O MONIT
```

Where:

- **O MONIT**
  - Controls automatic node reactivation.

**Related Statements:** STATOPT

O RESET

The O RESET statement defines initial status of resources which are known to status monitor but unknown to the VTAM associated with the status monitor. Using this option, the initial status for these resources is RESET. Without this option, the initial status for these resources is NEVER ACTIVE.

Unless the O RESET statement is used, the SNA topology manager uses the status of NEVER ACTIVE continuously when resolving the status of multiple-owned resources, rather than ignoring this VTAM node’s perspective of the status. Since the VTAM node doesn’t know of the existence of this resource, using O RESET
O RESET

enables the SNA topology manager to resolve the status of multiple-owned resources in the same way as when a major node is inactivated and the last status reported is RESET.

Without this option, the SNA topology manager will resolve the status of multiple-owned resources as if the VTAM node knows of the existence of this resource but the resource has never been activated.

Code the O RESET option control statement in the DSICNM member. To implement changes to DSICNM, stop and restart the VMT task. If changing the member results in additional extents being used, stop and restart the NetView program.

The syntax for the O RESET statement is:

O RESET

Where:

O RESET
Controls the initial status of resources which are known to the status monitor.

O SECSTAT

The O SECSTAT statement forces the status monitor to run as a secondary network resource status monitor in communication with VTAM Version 3 Release 3. With this interface, the status monitor does not receive unsolicited messages.

Code the O SECSTAT statement in DSICNM. To implement changes to DSICNM, stop and restart the VMT task. If changing the member results in additional extents being used, stop and restart the NetView program.

The syntax for the O SECSTAT statement is:

O SECSTAT

Where:

O SECSTAT
Designates this status monitor to run as a secondary network resource status monitor.

Usage Notes:

- If you have two NetView programs active in the same host, specify which NetView program gets the network resource status updates from VTAM. Uncomment the O SECSTAT statement in DSICNM when the NetView program you are installing is not responsible for the network’s status. This situation can occur when console and network automation are performed by more than one NetView program.
If the NetView program you are installing is responsible for the network’s status, leave this statement commented out in DSICNM or delete it.

If you do not specify O SECSTAT, the first status monitor that you initialize receives the network’s status updates from VTAM. If you specify O SECSTAT in DSICNM for both NetView programs, neither one gets status updates from VTAM.

**O SENDMSG**

The O SENDMSG statement causes the status monitor to issue message CNM094I for each resource specified on the SENDMSG statement during status monitor initialization. Without this option, the message is only issued when the states of the specified resources change from the initial values known to the status monitor.

Code the O SENDMSG statement in DSICNM. To implement changes to DSICNM, stop and restart the xxxxxxVMT task. If changing the member results in additional extents being used, stop and restart the NetView program.

The syntax for the O SENDMSG statement is:

```
O SENDMSG
```

*Where:*

**O SENDMSG**

Controls the generation of CNM094I at initialization.

**Related Statements:** SENDMSG

---

**OPCLASS**

The OPCLASS statement is obsolete. It is used for migration purposes only. For more information, refer to the *Tivoli NetView for z/OS Installation: Migration Guide*.

**Related Statements:** CMDCLASS, KEYCLASS, VALCLASS

---

**OpDsPrefix**

The OpDsPrefix statement sets the common global variable OpDsPrefix to your operator data set prefix. This statement is coded in CNMSTYLE.

The syntax for the OpDsPrefix statement is:

```
OpDsPrefix
```

*Where:*

`prefix`

Specifies the data set prefix. The default naming convention is NETVIEW.OPDS.
Usage Notes:

- For more information on defining operator data sets, refer to the Tivoli NetView for z/OS Installation: Configuring Additional Components.
- To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE OPDSPREFIX command.

OPERATOR

When SECOPTS.OPERSEC is not specified as SAFDEF in CNMSTYLE or OPERSEC=SAFDEF is not specified on the REFRESH command, the OPERATOR statement identifies each operator who can log on to this NetView program or who can start a session with this NetView program from a NetView program in another domain. This statement is also used to define operator identifiers that can be started as automation tasks by the AUTOTASK command. The OPERATOR statement must come before its associated PROFILEN statements. You code this statement in DSIOFP.

You can dynamically add or delete operators by adding or deleting OPERATOR statements in DSIOFP and issuing the REFRESH OPERS command.

The syntax for the OPERATOR statement is:

OPERATOR

where

- opid Indicates the 1–8 character value that identifies an operator. Valid characters for the operator identifier are letters A–Z, the numbers 0–9, or the special characters number sign (#), at sign (@), or dollar sign ($). The identifier must begin in column 1. Each operator must have a unique operator identifier. Also, code an operator statement for each operator identifier you want to use for an automation task. Do not use the names of hardcopy logs, terminals, or task identifiers as operator identifiers. The following identifiers are reserved by the NetView program and cannot be used as operator identifiers:
  - ALL
  - DPR
  - DST
  - HCL
  - HCT
  - LOG
  - MNT
  - NNT
  - OPT
  - OST
  - PPT
  - SYSOP
  - TCT

Additionally, if the operator identifier is the same as the LU name (terminal), some command lists assume that the operator is an autotask and do not run.
PASSWORD=password
Indicates the 1–8 character operator password. You are required to code a password, but the password is ignored if you code SECOPTS.OPERSEC=MINIMAL in CNMSTYLE. The password is also ignored if you use this operator identifier when starting an autotask using the AUTOTASK command. For operator identifiers set up specifically for autotasks, use the password to identify the operator as such.

NOCHECK
Allows the NetView operator to log on without NetView verifying the password. This option is only valid when SECOPTS.OPERSEC=NETVPW is set in CNMSTYLE. This function assumes that the NetView logon installation exit performs password verification. In this case, message DWO354 is sent to the authorized receiver indicating that the operator has logged on and the password has not been verified by the NetView program. The advantage of this option over OPERSEC=MINIMAL is that the operator’s profile is used and HCL or initial commands specified is used. NOCHECK must be preceded by a comma.

Note: Prior to NetView Version 3, OPERSEC=NETVPW was VERIFY=NORMAL.

Related Statements: SECOPTS, PROFI LEN

OPERID

The OPERID statement enables you to define the encryption keys for each NetView 3270 management console operator. Define the encryption keys for each operator in member DSITCPRF, which is in DSIPRF.

The syntax for the OPERID statement follows:

OPERID

Where:

 OPERID
operid

operid
Specifies the NetView operator ID. Define a NetView operator ID in the same manner as an existing ID. As a security benefit, only operators defined in DSITCPRF can log on to NetView using a NetView 3270 management console.

 ANY_OTHER

You can specify ANY_OTHER as the operator ID for operators that are not otherwise defined in DSITCPRF. This provides universal access.

 NETCONV_IP

The NETCONV_IP entry in DSITCPRF defines the encryption keys for NETCONV sessions that use TCP/IP. This entry must be in the DSITCPRF member if you use the tserver tcpipkey command at the workstation to change the encryption keys. To change the encryption keys on the NETCONV sessions, task CNMTAMEL must be stopped and restarted if it is active. The
ANY_OTHER value is NOT used as a default setting for the NETCONV_IP setting. If NETCONV_IP is omitted, a predefined value is used for the encryption keys.

WEB_SERVER
The WEB_SERVER entry in DSITCPRF defines the encryption keys for HTTP server sessions. To change the encryption keys on the HTTP server sessions, task DSIWBTSK must be stopped and restarted if it is active. The ANY_OTHER value is not used for the WEB_SERVER statement.

key1
Specifies the encryption key for the data flowing from the NetView 3270 management console to NetView (command flow). The length of the key can be in the range of 1–8 characters. Using an 8-character key is recommended. Encryption keys can be mixed case, but default and disabled have special meaning. Choose random printable nonblank characters. The encryption keys should be coded as 8 character values, or 16 digits of hexadecimal digits. If the encryption key is 16 characters and all of the character values correspond to hexadecimal digits, the key is converted to an 8 byte hexadecimal string. For all other cases the key is expanded or truncated to an 8 character result.

Specifying default, all in lowercase, means NetView provides a default encryption key. The default key is the same for any session, but is not a published value. This provides a minimal level of encryption protection. If key1 is specified as default, key2 must also be specified as default.

Specifying disabled, all in lowercase, means no encryption. Use this for debugging session problems in low-risk networks. If key1 is specified as disabled, key2 must also be specified as disabled.

key2
Specifies the encryption key for the data flowing from NetView to the NetView 3270 management console. The length of the key can be in the range of 1–8 characters. Using an 8-character key is recommended. Encryption keys can be mixed case. Choose random printable nonblank characters.

Specify default, all in lowercase, means NetView provides a default encryption key. If key1 is specified as default, key2 must be specified as default.

Specify disabled, all in lowercase, for no encryption. If key1 is specified as disabled, key2 must be specified as disabled.

Usage Notes:
- The operator ID must be followed by a colon and any number of blanks.
- The DSITCPRF member should be placed in a secure (DSIPRF DD) library.
- If the NetView 3270 management console is launched from the NetView management console (NMC), both encryption keys must be defined as default.

ORIGNET
The ORIGNET statement defines the name of the network to which the LU, COS, and MODE statements apply. Code this statement in the members specified on the ALIASMEM statement. The sample member supplied with the NetView program that contains the ORIGNET statement is DSIALTAB. Changes to the member containing the MODE statements do not take effect until you stop and restart the ALIASAPL task.

The syntax for the ORIGNET statement is:
ORIGNET

label ORIGNET networkname

Where:

label
Indicates the optional label for the ORIGNET statement. This label identifies the statement for any related error messages.

networkname
Indicates a 1–8 character name of the network requesting the translation.

Related Statements: ALIASMEM, COS, LU, MODE

PARMSYN

The PARMSYN statement defines a synonym for a keyword or value of commands defined to the NetView program. Code this statement in DSICMD. Stop and restart the NetView program to implement the changes.

When you use help, only the original keyword or value is used in the help information.

All PARMSYN statements for a CMDMDL must follow the CMDMDL and precede the CMDSYN statements.

The syntax for the PARMSYN statement is:

PARMSYN

label PARMSYN regularvalue,additionalvalue

Where:

label
Indicates the optional label for the PARMSYN statement. This label identifies the statement for any related error messages.

regularvalue
Indicates the original command operand name.

additionalvalue
Indicates a new 1–8 character operand name. After the NetView program is recycled, it will accept either the old or the new name.

Usage Notes:
• Do not use PARMSYN for VTAM, MVS, and terminal access facility (TAF) LU1 commands.
• See the DSICMD sample for the Tivoli-defined NetView program synonyms and then customize DSICMD for your NetView program use.
The PARTNER statement specifies whether communications with an LU should use persistent conversations. Use this statement only for high performance sessions.

This statement is valid only in DSIHINIT, and it overrides the default value set up by the DSITINIT statement.

The syntax for the PARTNER statement is:

```
PARTNER

label
PARTNER NAME=partname

NETID=netid

PERSIST=YES

PERSIST=NO
```

Where:

* **label**
  - Indicates the optional label for the PARTNER statement. This label identifies the statement for any related error messages.

* **NAME=partname**
  - Specifies a 1–8 character partner name (an LU or CP name) to which this statement applies. This is a required keyword.

* **NETID=netid**
  - Specifies a 1–8 character network name containing the partner LU (command processor). This keyword is optional and defaults to the local NETID.

* **PERSIST=YES | NO**
  - Defines whether all conversations between the NetView program and the partner node are persistent or nonpersistent.
    - **YES**
      - Is the default and indicates that all conversations are persistent. The conversations remain active regardless of the time elapsed for SEND requests between this NetView program and the partner node. YES is the default.
    - **NO**
      - Indicates that all conversations are nonpersistent. The conversations are deallocated if the time elapsed for SEND requests between this NetView program and the partner node reaches the nonpersistent time-out interval specified in DSICTMOD.

Usage Notes:

- If two nodes in two networks have the same LU name, VTAM may sometimes locate one or the other, depending on the active configuration.
- The PARTNER statement overrides the default value of PERSIST that is defined for all partner nodes in the DSTINIT statement.
The PCLASS statement controls the measurement operands used by the response time monitor. These measurement operands include:

- The time boundaries between the response time counters
- The response time definition used for a session
- The response time objective used for a session
- Whether you can display the response time of the last transaction for the session

Two statements control the response time monitor. The first, PCLASS, is used to define performance classes. The second, MAPSESS, is used to map each session into one of the defined performance classes. A performance class determines the response time measurement operands for the sessions mapped into that performance class.

If you code PCLASS and MAPSESS statements, create a member and put them in that member. The name of this new member must match the name coded on the NLDM.PERFMEM statement in CNMSTYLE.

Code one PCLASS definition statement for each performance class you define. These PCLASS statements are the first statements that appear in the member named on the NLDM.PERFMEM statement in CNMSTYLE.

If you do not code any of the statements discussed in this section, the following defaults are used for all sessions:

- The boundaries between the counters are at 1, 2, 5, and 10 seconds.
- Response time is defined as *time to first character*.
- You cannot display your own response time.
- No response time objective is defined.

If the defaults are acceptable, you can omit this statement. Do not code the NLDM.PERFMEM statement in CNMSTYLE.

The syntax for the PCLASS statement is:

```
PCLASS
```

Where:

- `cname` is the name of the performance class you define. This name is used on one or
more MAPSESS statements as the value of \textit{cname} in the PCLASS operand. The name starts in column 1 of the statement. The name is required.

\textbf{BOUNDS=}(1,2,5,10) | (a,b,c,d)

Lists the upper time limit on each of the first four response time counters. You can specify these times in minutes, seconds, and tenths of seconds (mm:ss.s), in seconds only (ss or ss.s), in tenths of seconds only (.s), or in minutes only (mm:). The maximum time you can specify is 1800 seconds. Values of 60 seconds or more are converted to minutes for display.

The first counter is incremented when response time is less-than or equal-to \textit{a}, the second counter is incremented when response time is greater-than \textit{a} but less-than or equal-to \textit{b}, and so on. The fifth response time counter is incremented when response time is greater-than \textit{d}.

If you specify a BOUNDS operand with less than four boundaries coded, the maximum time allowed is used for the omitted boundaries. If the BOUNDS operand is omitted, the default is BOUNDS=(1,2,5,10).

\textbf{RTDEF=FIRST|CDEB|KEYBD|LAST}

Defines how response time is measured. The response time is measured from the activation of a transaction, such as when the ENTER key is pressed, until a response is received. This operand enables you to define received in one of the following ways:

- **FIRST** Means the first character of the reply from the host arrived at the user’s terminal. If you omit RTDEF, FIRST is the default.
- **CDEB** Means an SNA Change Direction or an SNA End Bracket was received at the user’s terminal.
- **KEYBD** Means the user’s terminal keyboard is unlocked.
- **LAST** Means the last character of the reply from the host was received at the user’s terminal.

\textbf{DSPLYLOC=NO|YES}

Controls the display of response time.

- **NO** No response is displayed. This is the default, which is set by the RTMDISP initialization statement.
- **YES** Allows the terminal user to display the response time of the last transaction at the terminal.

\textbf{OBJTIME=00:00.0 | mm:ss.s}

Indicates the time threshold of the performance objective. It can be entered as minutes, seconds, and tenths of seconds (mm:ss.s), in seconds only (ss or ss.s), in tenths of seconds only (.s), or in minutes only (mm:). The maximum time is 1800 seconds. Values of 60 seconds or more are converted to minutes for display. The default is zero (0).

If OBJTIME is 0, no response time objective data is displayed for sessions mapped to this PCLASS statement. See \textit{MAPSESS} on page 76 to determine how sessions are mapped to a PCLASS statement.

If OBJTIME is not equal to one of the values of the BOUNDS operand, it is rounded off to the nearest BOUNDS value.

A response time objective includes both a time threshold and a percentage. For example, for a particular terminal you might want response times to be less than 5 seconds (the threshold) for 80% (the percentage) of the activity.
OBJPCT=01ppp
Indicates the percentage portion of the performance objective where:

0 Indicates the default value.

ppp Indicates the percentage of transactions that should take less time than the time specified by OBJTIME.

Usage Notes:
- If a response time monitor is monitoring an LU-LU session and the PCLASS statement defining the session uses an RTDEF value not supported by the response time monitor, the PCLASS statement is ignored and the session is mapped to another class. If no other classes are defined or match this class, the default performance class is used.
- If OBJPCT is 0, no response time objective data is displayed for sessions mapped to this PCLASS statement. See the MAPSESS statement to determine how sessions are mapped to a PCLASS statement.

Related Statements: NLDM.PERFMEM, MAPSESS

POLICY
The POLICY statement loads definitions in the Policy Repository when the NetView program is initialized. This statement is coded in CNMSTYLE.

The syntax for the POLICY statement is:

POLICY.name=policy_file

Where:

name Specifies any set of characters as defined by each policy application. The NetView program provides the following policy files:

POLICY.AON = EZLCFG01 for AON
POLICY.GRAPHICS = DUIPOLCY for NMC status

policy_file Specifies a file name within DSIPARM that contains policy definitions.

Usage Notes:
- You can have one or more POLICY statements, depending on your needs.

PORT
The PORT statement defines the port for the specified TCP/IP address space on which NetView waits for connection requests. This statement specifies the NetView that will receive a connection request.

Note: The PORT statement is used in the following samples:

- DSIILGCF
PORT

- DSIREXCF
- DSIRSHCF
- DSIRTTTD
- DSITCPCF
- DSIWBMEM
- DUIFPMEM

The syntax for the PORT statement is:

```
PORT

PORT = number

Where:

number

Specifies the port number in the range of 1–65535.

Usage Note: For the NetView Web Server, a single blank must precede and follow the equal sign in the PORT statement.
```

PREFIX

The PREFIX statement enables you to define the information that is displayed with a message and the order in which the information is displayed on the command facility panel. The sample member containing the PREFIX statement is CNMSCNFT.

Each PREFIX statement defines a single column of the prefix, including the column tag and a single blank separator. You can code the PREFIX statement as many as 16 times.

The syntax of the PREFIX statement is:

```
PREFIX

PREFIX fieldname (n1:n2) fieldlist COLTAG='colid'

Where:

fieldname

Defines the content of a prefix field. Operands that can be used in fieldname are:

- TIME Displays the time in the format specified by the TIME statement.
- DATE Displays the date in the format specified by the DATE statement.
- DOMID Displays the domain name.
SENDER
Displays the originator of the message.

IFRAUSRC
Displays characters from the IFRAUSRC data field.

IFRAUSRB
Displays two bytes of user bits formatted in hexadecimal from the IFRAUSRB field.

JOBID
Displays the job identifier.

JOBNAME
Displays the job name.

NMFLAGS
Displays the NetView message flags:
- P Specifies that the message is from the primary program operator interface task (PPT).
- % Specifies that the message is an unsolicited primary message.
- * Specifies that the message is an unsolicited secondary message.
- + Specifies that the message is a solicited copy message.

NMTYPE
Displays the NetView message type.

SRCNETID
Displays the network identifier from the source object.

SRCNAU
Displays the network addressable unit (NAU) from the source object.

SYSID
Displays the system identifier.

SESSID
Displays the terminal access facility session name or the SAF name associated with the PPI sender of this message.

\( n1:n2 \) Denotes the first and last characters of the string to be displayed. For example, 5:12, specifies that an 8-character string is taken from the \textit{fieldname} beginning at the fifth character.

\textit{fieldlist} Allows up to four mutually exclusive operands to be combined into one field. The operands are separated by logical or (\texttt{|}) signs. Only the first nonblank operand in the list that exists for the message is displayed. You can specify any of the operands listed under \textit{fieldname}. If you do not specify the \texttt{COLTAG} keyword with \textit{fieldlist}, the column tag from the first field listed is used as the default.

\texttt{COLTAG='colid'} Enables you to define the column identifier. The default column identifiers are:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Column Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOMID</td>
<td>DOMID</td>
</tr>
<tr>
<td>SENDER</td>
<td>SENDER</td>
</tr>
</tbody>
</table>
### PREFIX

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFRAUSRC</td>
<td>IFRAUSRC</td>
</tr>
<tr>
<td>IFRAUSRBB</td>
<td>IFRAUSRBB</td>
</tr>
<tr>
<td>JOBID</td>
<td>JOBID</td>
</tr>
<tr>
<td>JOBNAME</td>
<td>JOBNAME</td>
</tr>
<tr>
<td>NMFLAGS</td>
<td>P%</td>
</tr>
<tr>
<td>NTYPE</td>
<td>T</td>
</tr>
<tr>
<td>SRCNETID</td>
<td>SRCNETID</td>
</tr>
<tr>
<td>SRCNAU</td>
<td>SRCNAU</td>
</tr>
<tr>
<td>SYSID</td>
<td>SYSID</td>
</tr>
<tr>
<td>SESSID</td>
<td>SESSID</td>
</tr>
</tbody>
</table>

**Usage Notes:**
- To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- Each PREFIX statement enables you to choose the information or information options that are in a field, and the identifier of the column. The order of the PREFIX statements determines the order of the information on the panel. The NetView-supplied defaults for the PREFIX statement are NMTYPE, DOMID, and NMFLAGS (in that order). These defaults are used if you do not code the PREFIX or NOPREFIX statements.
- Each column expands to the size of the information field or the size of the identifier, whichever is greater. The total length of all of the columns you specify cannot exceed 75 characters.

**Related Statements:** ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

### PROFILE

The PROFILE statement defines the profile name to the system. PROFILE must be the first statement in each profile definition. Code this statement in a member specified by a PROFILEN statement associated with the operator. See "OPERATOR" on page 150 and "PROFILEN" on page 161 to determine how a PROFILEN statement is associated with an operator. Profiles are not used when OPERSEC=SAFDEF. Examples of sample members supplied with the NetView program are DSIPROFA and DSIPROFB.

The syntax for the PROFILE statement is:

```
PROFILE
    profilename PROFILE [HCL=hclname] [CONSNAME=consname] [IC=text]
```

**Where:**
- `profilename`
  Indicates the name of the member that contains the profile. This name must begin in column 1.
HCL=hostname
Indicates the name of the hardcopy printer that is automatically started when
this operator logs on. Define this name in the VTAM definition and in the
HARDCOPY statement in CNMSTYLE. HCL is an optional operand. The IC
classification, when specified, must always be specified as the last keyword.
Although each operator can be assigned to only one hardcopy printer, several
operators can share the same printer. However, if too many operators share the
same hardcopy printer, messages for that device can accumulate and messages
might not be printed for some time after they are received.

CONSNAME=consname
Indicates the default extended console name for operators using this profile.
This default console name is used when the operator does not specify a
console name using the GETCONID or SETCONID command. It is also the
console name used when you issue the MVS command and have not
previously obtained an extended console. The console name must be a 2- to
8-character value, as required by MVS. Valid characters for console names are
A–Z, 0–9, @, #, or $. The first character of the console name must be alphabetic
(A–Z) or one of the following special characters: @, #, or $. For more
information on the implications of specifying CONSNAME, refer to the
GETCONID and SETCONID command in the NetView online help. The IC
classification, when specified, must always be specified as the last keyword.

IC=text
Indicates the command or command list that is run immediately after a
successful log on. All the text on the remainder of the statement through
column 71 is treated as the text of the command, including embedded blanks
and commas. No enclosing quotation marks are allowed. If text specifies a
command, the command must be defined by a CMDMDL statement as regular
(R), both (B), or high (H). The IC classification, when specified, must always be
specified as the last keyword.

Related Statements: HARDCOPY, OPERATOR, PROFILEN

PROFILEN
The PROFILEN statement associates the name of a particular profile or list of
profiles with an operator identification. Code PROFILEN as often as necessary to
ensure that all the possible profile names are associated with a particular operator
identification. An OPERATOR statement must precede each PROFILEN statement
or group of statements. You code this statement in DSIOPF. Profiles are not used
when SECOPTS.OPERSEC=SAFDEF is specified in CNMSTYLE.

The syntax for the PROFILEN statement is:

PROFILEN

Where:
PROFILEN

*label*

Indicates the optional label for the PROFILEN statement. This label identifies the statement in any related error messages.

*profilename [,...]*

Indicates the profile name to be associated with the operator identification defined in the preceding OPERATOR statement. The profile name is a 1–8 character name that matches the *profilename* given on a PROFILE statement in a profile member. The first name listed in the first PROFILEN statement is used by default if an operator does not specify a *profilename* in the log on request. Note that multiple profile names must be separated by commas.

**Related Statements:** OPERATOR, PROFILE

PROTOCOL

The PROTOCOL statement defines the type of remote IP command server being used.

**Note:** The PROTOCOL statement is used in the following samples:
- DSIREXCF
- DSIRSHCF

The syntax for the PROTOCOL statement is:

```
PROTOCOL = REXEC
```

**Where:**

REXEC

Sspecifies that the DSIRXEXC task is an REXEC server.

RSH

Sspecifies that the DSIRSH task is an RSH server.

**Usage Notes:**
- There is no default value if the statement is not present.

PUCOUNT

Use the PUCOUNT statement to determine the size of the registration table used for programmable network access (PNA) support. The value you specify is the expected number of registered physical units (PUs), which is a PNA plus all PUs attached to it, for the domain. The value you specify is rounded up to the next largest prime number. A precise number is not necessary, but the value does have performance implications. Specifying too small a value causes additional overhead when registration records are added to the table. Specifying too large a value causes additional unused memory to be allocated for the registration table.

Code the PUCOUNT statement in the member specified by the MEM keyword on the DSIROVS TASK statement. The sample member supplied with the NetView program is DSIROVSI. To implement changes to DSIPARM, stop and restart the task. If changing the member results in additional extents being used, stop and restart the NetView program.
The syntax for the PUCOUNT statement is:

```
PUCOUNT
```

Where:

2003 | nnnnn
Indicates the expected number of registered PUs. The value can be a decimal number in the range of 3–32749. The default is 2003.

**RCMRODMUser**

The RCMRODMUser statement specifies the RODM application identifier as a global variable. The RODM application identifier is used by the RODM Collection Manager (RCM) to sign on to RODM. This statement is coded in CNMSTYLE.

The syntax for the RCMRODMUser statement is:

```
RCMRODMUser
```

Where:

applid
Indicates a 1–5 character RODM application identifier.

Usage Notes:

- Set the applid to an SAF authorized user ID with the authority to sign on to RODM as well as the authority to create and change RODM data.

**REP**

The REP statement enables you to define a RODM to the RODM access and control task. You can define up to 64 RODMs to the RODM access and control task.

The syntax for the REP statement is:

```
REP
```

163
**Where:**

*rodmaname*

Specifies a valid RODM name. The name can be 1–8 characters.

**Note:** Use unique identifiers if you plan to access the same RODM from more than one NetView. The ID must be authorized in RACF to access the named RODM at the desired level for automation platform access.

**CONN=N|Y**

Specifies whether connection to RODM occurs automatically at DSIQTSK initialization.

- **N** Specifies that connection does not occur automatically. N is the default.
- **Y** Specifies that connection occurs automatically.

**AO=N|Y**

Specifies whether this RODM is the current run-time RODM. You can define only one RODM as the current runtime RODM.

- **N** Specifies that this RODM is not the current run-time RODM. N is the default.
- **Y** Specifies that this RODM is the current run-time RODM.

**T=120|seconds**

Specifies the time, in seconds, for which the NetView program waits while RODM is taking a checkpoint. The value can be in the range of 10–3600 seconds. The default is 120 seconds.

**CMD=command**

Specifies the name of a valid initialization command processor or command list. The command is invoked when one of the following conditions is met:

- Connection to RODM occurs.
  
  The command is invoked with the following parameters:
  
  ![Image](image)

  Where *rodmaname* is the name of the RODM. The command is executed at the first connection only and not on subsequent connection requests when a connection already exists.

- There is a run-time RODM change.

  The command is invoked with the following parameters:
  
  ![Image](image)

  Where *rodmaname* is the name of the RODM.

**ID=DSIQTSK|applid**

Specifies the user application ID used to access RODM.

- **DSIQTSK** Indicates the name of the DSIQTSK task. This is the default.

  ![Image](image)

- **applid** Specifies the ID of a valid user application. The length of this value cannot exceed 8 characters.
RETRY

The RETRY statement enables you to define the number of retry attempts and the interval of time (in seconds) between retries for an APPCCMD that has failed with a return code that allows a request retry.

The syntax for the RETRY statement is:

```
RETRY
```

Where:

ATTEMPTS
Specifies the number of retry attempts NetView makes for an APPCCMD after it has failed with a return code that allows a request retry. The default is 2.

Specifying 0 (zero) means no retry.

INTERVAL
Specifies how long NetView waits (in seconds) before an APPCCMD request is retried after the APPCCMD has failed with a return code that allows a request retry. The default is 0 (zero) which causes NetView to retry the request immediately.

Related Tasks: DSI6DST, DSIHPDST

REXEC

The REXEC statement defines TCP/IP definitions for the REXEC server (task DSIRXEXC, member DSIREXCF). Code this statement in CNMSTYLE.

The syntax for the REXEC statement is:

```
REXEC
```

Where:

TCPANAME = name
Defines the TCP/IP address space name.

PORT = port
Defines the port number on which DSIREXCF waits for connection requests.

   The default is 512.

SOCKETS = number
Specifies the maximum number of simultaneous users. The default is 50.
**RExec**

**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE REXEC command.

---

**RMTINIT**

The RMTINIT statement in CNMSTYLE enables supported modes of communication (SNA LU 6.2 and TCP/IP) and provides additional information for communication.

The syntax for the RMTINIT statement is:

```
RMTINIT
```

Where:

**RMTINIT.SECOPT**

RMTINIT.SECOPT defines the type of security you want to use for RMTCMD and ENDTASK processing. The security specified allows you to control which operators in which domains are allowed to establish and end RMTCMD sessions with autotasks in your domain. You can use the REFRESH command to change RMTCMD security while the NetView program and the RMTCMD data services task (DSIUDST) are running. Possible values are as follows:

* **NONE**
  - Specifies that no security check is made. Any NetView operator from any network and domain can start or end any autotask using RMTCMD or ENDTASK. *NONE* is specified in the sample CNMSTYLE member shipped with NetView.

* **SAF**
  - Specifies that the RMTOPS class of the SAF product is to be called for authorization checking of the initial RMTCMD request and ENDTASK requests. If a RMTCMD security table was previously in use, the storage for the table is reclaimed.

  **tblname**
  - Specifies the 1 to 8 character name of the table used for RMTCMD security verification. For additional information on how to build the table, see the [Tivoli NetView for z/OS Security Reference](#). The NetView-supplied sample for the RMTCMD security validation table is DSISECUR.

  **RMTINIT.SAFrefresh=YES | NO**
  - Specifies whether the security option can be dynamically changed from TABLE or NONE to SAF by the REFRESH command.
YES  Specifies that the security level can be changed to SAF with the
REFRESH command. This is the default.

NO  Specifies that the security level cannot be changed to SAF with the
REFRESH command. If you specify NO, the NetView program does
not try to establish a security environment that allows the use of the
RACROUTE macro for RMTCMD authorization checking.
SAFREFSH=NO is not valid when you specify the security level as
SAF.

RMTINIT.SNA
  Specifies whether SNA LU 6.2 remote operations is enabled.

RMTINIT.IP
  Specifies whether TCP/IP remote operations is enabled.

RMTINIT.TCPNAME
  Specifies the name of the TCP/IP stack to be used for remote operations. This
statement is read only if RMTINIT.IP is yes.

RMTINIT.PORT
  Specifies the port number to which DSIUDST will bind and receives remote
operations connection requests. This statement is read only if RMTINIT.IP is
yes.
    portnum
      A number from 0 to 65535. Specifying 0 requests that TCP/IP assign a port
number.

RMTINIT.SOCKETS
  Provides a maximum number of sockets to be available to DSIUDST for
supporting TCP/IP remote operations. This keyword is read only if
RMTINIT.IP is Yes.
    socketnum
      A number from 50 to 2000.

RMTSEC

The RMTSEC statement authorizes or blocks remote operators from using the
RMTCMD or ENDTASK commands to start or stop autotasks on a distributed
NetView program. The distributed NetView host adds RMTSEC statements in the
table used for RMTCMD security verification for remote operators, nodes, or LUs
that operators want to authorize or block. The NetView-supplied sample for the
RMTCMD security validation table is DSISECUR.

For more information on using this table for RMTCMD and ENDTASK security,
see Tivoli NetView for z/OS Security Reference.

The syntax for the RMTSEC statement is:

```
RMTSEC
  BLOCK
  PASS
  TARGOP=autotaskid
  T=* autotaskid
```

Where:

**BLOCK|B**
Specifies that the request should be denied.

**PASS|P**
Specifies that the request should be accepted and the autotask started or stopped.

**TARGOP|T=autotaskid|*|\**
Specifies the operator ID at the distributed NetView program that is to be started or stopped as a distributed autotask. If you specify this operand, you must specify RMTOP. If you do not specify this operand, you cannot specify RMTOP.

* Specifies any. You can also use the asterisk at the end of a string to match all names beginning with the specified string. For example, CNM* means that any name for that field beginning with CNM produces a match.

\ (EBCDIC X'E0' for backslash)
You can use a backslash (\) to indicate that TARGOP and RMTOP must have the same value. You can use a backslash only on TARGOP and RMTOP. Omitting both TARGOP and RMTOP from the RMTSEC statement has the same effect as coding the backslash and might be preferable when using a keyboard other than an American English model.

**NET|N=netid|**
Specifies the network identifier of the remote operator that issued the incoming RMTCMD or ENDTASK request.

**LU|L=luname|**
Specifies the domain identifier of the remote operator that issued the incoming RMTCMD or ENDTASK request.

**RMTOP|R=remoteopid|*|\**
Specifies the user ID of the remote operator issuing the incoming RMTCMD or ENDTASK request. If you specify this operand, you must specify TARGOP. If you do not specify this operand, you cannot specify TARGOP.

**CMD|C=RMTCMD|ENDTASK|**
Specifies whether you want to control starting or ending the autotask by using the RMTCMD or ENDTASK commands, respectively.

**DOMAIN|D=domainname|**
Specifies the domain identifier of the remote operator that issued the incoming RMTCMD or ENDTASK request. This keyword can not be specified with LU or L (nor can D be specified with DOMAIN) but it uses the same values and serves the same function as LU or L.
Usage Notes:

- You can use a question mark (?) to replace one character as a wildcard character. For example:

  \texttt{RMTSEC P TARGOP=O??R* NET=NETA LU=C???A RMTOP=NETOP? CMD=*}

Results in the following:
- \texttt{TARGOP=O??R*} will find OPER1 and OPER22, but it will not find OPR1.
- \texttt{NET=NETA} will find only NETA.
- \texttt{LU=C???A} will find C123A and CIBMA, but it will not find C34A or CIBMAA.
- \texttt{RMTOP=NETOP?} will find NETOP1 and NETOPA, but it will not find NETOP or NETOP22.
- \texttt{CMD=*} will find RMTCMD or ENDTASK.

- Use the REFRESH command to read in new definitions while the NetView program is running. Refer to the NetView online help for more information about the REFRESH command.

---

\textbf{RMTSYN}

The \texttt{RMTSYN} statement in DSIPARM member CNMSTYLE specifies the protocol used for communications by the PIPE ROUTE stage as well as by the RMTCMD and LABEL commands.

Syntax for the RMTSYN statement:

\begin{verbatim}
RMTSYN
  netid. domain = domain_name /port_number
  IP_address /port_number
  SNA
\end{verbatim}

Where:

- \texttt{netid}
  - Specifies the 1–8 character name of the network containing the \texttt{domain}.

- \texttt{domain}
  - Specifies the 1–5 character name of the remote NetView domain.

- \texttt{domain_name}
  - Specifies a standard TCP/IP host and domain name for the site of the remote NetView. When communicating using TCP/IP, the domain name can be followed by a port number. Domain names are resolved to IP addresses after the main part of initialization.

- \texttt{port_number}
  - Specifies the port number. A slash (/) separates the \texttt{port_name} from the \texttt{domain_name} or \texttt{IP_address}. The default port number is 4022.

- \texttt{IP_address}
  - Specifies the TCP/IP address in dotted decimal notation that is to be used to communicate with the specified \texttt{domain}. The address can be followed by a port number.
RMTSYN

SNA

Specifies to use SNA LU 6.2 to communicate with the specified domain. This is the default.

Usage Notes:

- An example RMTSYN statement follows using a domain name:
  `RMTSYN.USIBMNT.NTV7E = mmpi102.raleigh.ibm.com/4022 // resolve later` 
- An example RMTSYN statement follows using an IP address and a default port:
  `RMTSYN.USIBMNT.NTV6D = 9.67.50.22 // IP address & default port` 
- An example RMTSYN statement follows for a session using LU 6.2:
  `RMTSYN.USIBMNT.NTVAA = SNA // use LU6.2 for this domain`

RODMname

The RODMname statement specifies the RODM name as a global variable. This statement is coded in CNMSTYLE.

The syntax for the RODMname statement is:

```
RODMname

Where:
RODM_name
  Indicates a 1–5 character RODM name.

Usage Notes:
- This statement is ignored if you are not using RODM.
- If you are using GMFHS, specify the same RODM_name in DUIGINIT that you specify on this statement.
- If you set the system symbolic &CNMRODM in SYS1.PARMLIB member IEASYMxx, do not update this statement.
```

RRD

The RRD statement shows the domains that can be used to start NNT cross-domain sessions using the START TASK command. The RRD statements tells the NetView program the domains that contain particular cross-domain resources. This statement is coded in CNMSTYLE.

The syntax for the RRD statement is:

Where:
domainid
Indicates the 1–5 character NetView program identifier of the domain with which this NetView program can start an NNT cross-domain session. The value of domainid should match the NetView program identifier in the other domain.

For interconnected networks, all NetView systems in session with each other must have unique identifiers that cannot be alias names.

nodeName [,...]
Specifies the node name of a resource. If you code this node name on an RRD statement, you do not have to explicitly state the domain identifier when you route VTAM commands.

The SSCP in the domain specified by domainid must know the node name because the NetView program sends routed commands directly to that domain. These commands do not undergo any alias name translation by the sending SSCP.

For interconnected networks, the alias name translation function allows the use of duplicate resource names in different networks. To avoid conflicts, you must define each node name only once within the set of RRD statements. To refer to a cross-domain resource with a duplicate name, state the appropriate domain name in the command.

* Specifies this domain, without any node names under it.

Usage Notes:
- Conversations with distributed autotasks started with the RMTCMD command are not affected by the RRD statement.

RSH

The RSH statement defines TCP/IP definitions for the RSH server (task DSIRSH, member DSIRSHCF). Code this statement in CNMSTYLE.

The syntax for the RSH statement is:

RSH

Where:
PORT = port
Defines the port number on which DSIRSH waits for connection requests. The default is 514.
RSH

SOCKETS = number

Specifies the maximum number of simultaneous users. The default is 50.

Usage Notes:

• RSH uses TCPname.
• To implement definition changes while the NetView program is active, change
  the value in CNMSTYLE, and enter the RESTYLE RSH command.

RTT

The RTT statement defines TCP/IP definitions for the alert receiver (task DSIRTTR,
member DSIRTTTD). Code this statement in CNMSTYLE.

The syntax for the RTT statement is:

RTT

Where:

TCPANAME = name

Defines the TCP/IP address space name.

PORT = port

Defines the port number on which DSIRTTR waits for connection requests. The
default is 4021.

SOCKETS = number

Specifies the maximum number of simultaneous requests. The default is 50.

Usage Note: To implement definition changes while the NetView program is
active, change the value in CNMSTYLE, and enter the RESTYLE RTT command.

SECOPTS.AUTHCHK

The SECOPTS.AUTHCHK statement specifies the user ID that is to be used when
verifying command authorization. For specific information about how the
SOURCEID and TARGETID are determined, refer to the Tivoli NetView for z/OS
Security Reference. This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.AUTHCHK statement is:

Where:

SOURCEID

Specifies to check the authority of the original issuer or the ID closest to the
original issuer.
Access failure messages display the source issuer of the command. SOURCEID is the default.

**TARGETID**

Specifies to check the authority of the task under which the command runs.

### SECOPTS.CMDAUTH

The SECOPTS.CMDAUTH statement defines the method used by NetView to protect command usage. For a list of commands with keywords and values that can be protected, see [Tivoli NetView for z/OS Security Reference](#). This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.CMDAUTH statement is:

```
SECOPTS.CMDAUTH= SAF. FAIL
  PASS
  tblname
  TABLE. tblname
  SCOPE. tblname
```

**Where:**

- **SAF.**
  - option
    - Specifies that the NetView program performs command authorization checking using an SAF security product. The commands you want to protect should be defined in the NETCMDS class in the security product. The operators that you want to give access to the commands can be permitted to use the resource names that represent commands, keywords, and values. For more information, refer to the [Tivoli NetView for z/OS Security Reference](#).

If during NetView initialization, the NETCMDS class is not active or the security product is not active and a backup command authorization table is not specified, an error message is issued and all commands are permitted.

For immediate commands, authority checking is not performed by SAF. These commands should be protected using the backup command authorization table.

Include one of the following options when you specify SAF:

- **FAIL**
  - Specifies that users should not be allowed to issue any commands when the SAF product cannot make a security decision.

- **PASS**
  - Specifies that users should be allowed to issue all commands when the SAF product cannot make a security decision.

- **tblname**
  - Specifies the 1–8 character name of the backup command authorization table used when SAF checking cannot be performed for the command being issued. This can occur when:
    - The command is an immediate command.
    - There is no resource name defined in the NETCMDS class which protects or authorizes this command.
    - The NETCMDS class is not active.
    - The security product is not active.
SECOPTS.CMDAUTH

The table name is a member name in a DSIPARM data set. For a sample backup command authorization table, refer to CNMSBAK1. For more information on how to build the table, see the Tivoli NetView for z/OS Security Reference.

The tblname and PASS|FAIL options are mutually exclusive. When both are specified, the tblname option is used.

If tblname contains syntax errors, messages are issued. NetView continues with no backup table.

If a valid backup command authorization table is not specified, immediate commands are not checked and pass authorization.

TABLE.tblname

Specifies the 1–8 character name of the command authorization table. This name is a member name in a DSIPARM data set. For a sample command authorization table, refer to CNMSCAT2. For more information on how to build the table, see the Tivoli NetView for z/OS Security Reference.

If the command authorization table contains syntax errors, or is not found, messages are issued and all commands are permitted.

SCOPE.tblname

The NetView program supports the SCOPE option in migration mode only. Scope of commands security definitions (CMDCLASS, KEYCLASS, VALCLASS statements in DSICMD, with matching OPCLASS statements) are converted into equivalent command authorization table statements by the SECMIGR command. The converted table is written to the first DSIPARM data set and is put into effect.

Usage Notes:

- SECOPTS.CMDAUTH is ignored when SECOPTS.OPERSEC=MINIMAL. With other SECOPTS.OPERSEC settings you can issue the REFRESH command to change the method used for command authorization.
- If the CMDMDL statement for a command specifies SEC=BY, no authority checking is done for that command. Command authorization checking for automation table commands can also be bypassed by specifying AUTOSEC=BYPASS on the DEFAULTS command. For more information, see the Tivoli NetView for z/OS Security Reference.
- Scope of commands is supported in migration mode only. Any CMDCLASS, KEYCLASS, or VALCLASS statements in DSICMD, as well as any OPCLASS statements in operator profiles or the NETVIEW segment in the SAF product are converted into equivalent command authorization table statements by the SECMIGR command.
- Table 11 shows the relationships between CMDAUTH keywords and SECTOPS.AUTHCHK specifications.

Table 11. Interrelationships between SECOPTS.CMDAUTH Keyword Values

<table>
<thead>
<tr>
<th>IF CMDAUTH=</th>
<th>TABLE</th>
<th>SAE:tblname</th>
<th>SAE:PASS</th>
<th>FAIL</th>
<th>SECOPTS.AUTHCHK=</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE</td>
<td>Required</td>
<td>Ignored</td>
<td>Ignored</td>
<td>Is valid</td>
<td>TARGETID</td>
</tr>
</tbody>
</table>
Table 11. Interrelationships between SECOPTS.CMDAUTH Keyword Values (continued)

<table>
<thead>
<tr>
<th>SAF</th>
<th>Ignored</th>
<th>Is valid</th>
<th>Is valid, defaults to PASS, and is ignored if tblname is specified</th>
<th>Is valid</th>
<th>Default</th>
</tr>
</thead>
</table>

* Table 12 shows how the SECOPTS.CMDAUTH and related options are used to specify how operator command authority checking is to be performed.

Table 12. Protecting Commands Executed in NetView

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Restrictions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOPTS.CMDAUTH = TABLE</td>
<td>Command authorization is based on the specified table.</td>
<td></td>
</tr>
<tr>
<td>SECOPTS.CMDAUTH = SAF</td>
<td>MVS RACF Version 2 or equivalent SAF security product is required. Cannot be specified if SECOPTS.OPERSEC is MINIMAL, NETVPW, or SAFPW.</td>
<td>Command authorization using the NETCMDS class of an SAF product. Immediate commands are not checked in the NETCMDS class, but a backup command authorization table can be used for this purpose.</td>
</tr>
<tr>
<td>SECOPTS.CMDAUTH = SAF.tblname</td>
<td>Specifies the backup table to be used for immediate commands and when the SAF product cannot make a security decision. This can occur when: - There is no resource name defined in the NETCMDS class which protects or authorizes this command - The NETCMDS class is not active - The security product is not active.</td>
<td></td>
</tr>
<tr>
<td>SECOPTS.CMDAUTH = SAF.PASS</td>
<td>FAIL</td>
<td>Defaults to PASS</td>
</tr>
</tbody>
</table>

* For more information, refer to Tivoli NetView for z/OS Security Reference.

SECOPTS.OPERSEC

The SECOPTS.OPERSEC statement defines the method used to allow users to log on to the NetView program. This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.OPERSEC statement is:

```
SECOPTS.OPERSEC=MINIMAL, NETVPW, SAFPW, SAFCHECK, SAFDEF
```

Where:

**MINIMAL**

Specifies that NetView operators are defined by a list of operator identifiers in
SECOPTS.OPERSEC

DSIOPF. There is no password validation. The logon profile is not used and logon operands specified on the logon menu are ignored.

Other SECOPTS keywords are ignored when SECOPTS.OPERSEC = MINIMAL is specified.

NETVPW

Specifies that NetView operators are defined by a list of operator identifiers in DSIOPF. The identification is validated with a password associated with the identifier in DSIOPF. The profile, read from DSIPRF at logon, contains information about what the operator is allowed to do, and limits commands and resources that the operator can use.

SECOPTS.CMDAUTH=SAF or SECOPTS.OPSPAN=SAF should not be specified when SECOPTS.OPERSEC=NETVPW.

SAFPW

Specifies that operator identification and password checking is performed using an SAF security product. The operator identifier must also be defined in DSIOPF, and other attributes given to the operator at logon are taken from the operator’s specified profile in DSIPRF.

Security access checks which occur when an operator tries to access a data set that is protected in the DATASET class of an SAF product or an MVS system command that is protected in the OPERCMDS class of an SAF product are checked against the authority of the NetView program and not the authority of the operator.

SECOPTS.CMDAUTH=SAF or SECOPTS.OPSPAN=SAF should not be specified when SECOPTS.OPERSEC=SAFPW. If SECOPTS.OPSPAN=SAF is specified, SECOPTS.OPSPAN=NETV is used.

SAFCHECK

Specifies that operator identification and password checking is performed using an SAF security product. Authority to log on as a NetView operator is controlled through the APPL class. The operator identifier must be authorized to the resource name in the APPL class which represents the NetView program.

This specification provides the same function as the ‘SAF CHECK AT TASK LEVEL’ byte in the NetView constants module DSICMOD in releases of the NetView program prior to Version 3. This setting specified security authorization against the authority of the operator when VERIFY=MAXIMUM was specified and is now ignored. If the DSICMOD byte is set to X’01’ an error message is issued and initialization continues.

SAFDEF

Specifies that operator identification and password checking is done using an SAF security product. Authority to log on as a NetView operator is controlled through the APPL class. The operator identifier must be authorized to the resource name in the APPL class which represents the NetView program.
The attributes given to the operator at logon are defined in the NETVIEW segment of the operator’s user profile in the SAF product. For more information, refer to [Tivoli NetView for z/OS Security Reference](#).

When SECOPTS.OPERSEC=SAFDEF is specified, any value for SECOPTS.CMDAUTH can be used.

When SECOPTS.OPERSEC=SAFDEF is specified, SECOPTS.OPSPAN=SAF is required. If SECOPTS.OPSPAN=NETV is specified, the SAF value is used, an error message is issued, and initialization continues.

**Usage Notes:**

- If SECOPTS.OPERSEC=SAFDEF, you do not need a DSIPRF specification in your NetView procedure. If you later want to issue a REFRESH command with OPERSEC specified as either NETVPW, SAFPW, or SAFCHECK, you must first dynamically allocate DSIPRF, if it is not in your NetView procedure.

- If SECOPTS.OPERSEC=SAFDEF, you do not need a DSIOPF member in DSIPARM. If you later want to issue a REFRESH command with OPERSEC specified as either NETVPW, SAFPW, or SAFCHECK, DSIPARM must contain a DSIOPF member for the REFRESH command to complete successfully.

- If SECOPTS.OPERSEC=SAFDEF, you can log on to NetView using a PassTicket rather than a password if you use the Network Security Program/Secure Logon Coordinator product (NetSP/SLC V1.2) with an SAF product which supports PassTickets, such as RACF.

  - Table 13 shows the relationships between SECOPTS.OPERSEC, SECOPTS.CMDAUTH, SECOPTS.OPSPAN, and SECOPTS.SPANAUTH specifications.

<table>
<thead>
<tr>
<th>IF SECOPTS. OPERSEC=</th>
<th>CMDAUTH</th>
<th>OPSPAN</th>
<th>SPANAUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TABLE</td>
<td>SAF</td>
<td>NETV</td>
</tr>
<tr>
<td>MINIMAL</td>
<td>Not valid</td>
<td>Not valid</td>
<td>Not valid</td>
</tr>
<tr>
<td>NETVPW</td>
<td>Is valid</td>
<td>Default</td>
<td>Forced to NETV</td>
</tr>
<tr>
<td>SAFPW</td>
<td>Is valid</td>
<td>Default</td>
<td>Forced to NETV</td>
</tr>
<tr>
<td>SAFCHECK</td>
<td>Is valid</td>
<td>Is valid</td>
<td>Default</td>
</tr>
<tr>
<td>SAFDEF</td>
<td>Is valid</td>
<td>Is valid</td>
<td>Forced to SAF</td>
</tr>
</tbody>
</table>

- Table 14 shows how the SECOPTS.OPERSEC and SECOPTS.OPSPAN statements are used to specify how operator verification and authority checking is to be performed.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Related Defaults</th>
<th>Restrictions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOPTS.OPERSEC = MINIMAL</td>
<td></td>
<td>- SECOPTS.SPANAUTH is ignored.</td>
<td>Logon profile ignored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SECOPTS.CMDAUTH is ignored.</td>
<td>Logon operands ignored</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SECTOPS.OPSPAN is ignored.</td>
<td>No password validation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cannot use REFRESH to change OPERSEC.</td>
<td>Operator must be defined in DSIOPF.</td>
</tr>
</tbody>
</table>
### Table 14. Defining and Verifying Operator Authority (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Related Defaults</th>
<th>Restrictions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOPTS.OPERSEC = NETVPW</td>
<td>– SECOPTS.OPSPAN default and only choice is NETV.</td>
<td>– Not valid when SECOPTS.CMDAUTH = SAF</td>
<td>– Password validated from DSIOPF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Not valid when SECOPTS.OPSPAN = SAF</td>
<td>– Operator must be defined in DSIOPF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– Profile read from DSIPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Password validated from DSIOPF.</td>
<td>– Operator must be defined in DSIOPF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Not valid when SECOPTS.OPSPAN = SAF</td>
<td>– Profile read from DSIPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Password verification using SAF product</td>
<td>– Profile read from DSIPRF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Operator must be defined in DSIOPF.</td>
<td>– NetView startup procedure name is used for any SAF calls made for NetView operators, such as to the DATASET class.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– The DSICTMOD byte for task-level checking is ignored.</td>
<td>– Replaces the use of DSICTMOD byte setting to indicate SAF checking at the task level.</td>
</tr>
<tr>
<td>SECTOPS.OPERSEC = SAFCHECK</td>
<td>– SECOPTS.OPSPAN default is NETV.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SECOPTS.OPERSEC = SAFDEF</td>
<td>– SECOPTS.OPSPAN default and only choice is SAF.</td>
<td>– RACF Version 2 or equivalent SAF security product is required</td>
<td>– Password verification using SAF product</td>
</tr>
<tr>
<td></td>
<td></td>
<td>– Cannot specify OPSPAN=NETV</td>
<td>– Operator logon authority using RACF APPL class</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– Operator attributes defined in NETVIEW segment of SAF product</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– NetView task userids are used for any SAF calls for NetView operators, such as to the DATASET class.</td>
</tr>
<tr>
<td>SECOPTS.OPSPAN = NETV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– NetView program authorizes operator to start spans through the SPAN and ISPAN statements.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– Operator must be defined in DSIOPF.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>– Profile read from DSIPRF.</td>
</tr>
</tbody>
</table>
### Table 14. Defining and Verifying Operator Authority (continued)

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Related Defaults</th>
<th>Restrictions</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECOPTS.OPSPAN = SAF</td>
<td>If SECOPTS.OPERSEC = SAFDEF, OPSPAN default is SAF.</td>
<td>- RACF Version 2 or equivalent SAF security product is required.                - SECOPTS.OPSPAN = SAF can only be specified when SECOPTS.OPERSEC = SAFCHECK or SECOPTS.OPERSEC = SAFDEF.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The NETSPAN class in the security product authorizes the operator to start spans.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Access level determines commands operator can issue against resources in a span.</td>
<td></td>
</tr>
</tbody>
</table>

- For more information, refer to [Tivoli NetView for z/OS Security Reference](#).

### SECOPTS.OPSPAN

The SECOPTS.OPSPAN statement defines the method for determining an operator’s authority to start spans of control. This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.OPSPAN statement is:

```
SECOPTS.OPSPAN = SAF
```

**Where:**

**NETV**

Specifies that the NetView program performs span checking based on the CTL specification and the SPAN and ISPAN statements in the operator profile in DSIPRF. Checking is performed when an operator issues a START SPAN command.

SECOPTS.OPSPAN=NETV is not supported when SECOPTS.OPERSEC=SAFDEF. If SECOPTS.OPSPAN=NETV is specified, SECOPTS.OPSPAN=SAF is used, an error message is issued, and initialization continues.

**SAF**

Specifies that the NetView program uses an SAF product to perform authorization checking when an operator issues a START SPAN command. The NETSPAN class in the security product contains a resource name to represent each span an operator can start and the operator must be permitted to use that span.

The operator can be given four levels of access to the resource which represents the span. The access level determines which commands the operator can issue against resources in the span. For example, READ access is required for VTAM DISPLAY commands, while update access is required for VTAM VARY commands. For more information, see the [Tivoli NetView for z/OS Security Reference](#).

When SECOPTS.OPERSEC is specified as NETVPW or SAFPW, SECOPTS.OPSPAN=NETV is required. If you have specified SECOPTS.OPSPAN as SAF in this case, the NetView program overrides this
specification and uses SECOPTS.OPSPAN=NETV. This override action causes an error message to be issued and initialization continues.

Usage Notes:

- SECOPTS.OPSPAN cannot be specified when SECOPTS.OPERSEC=MINIMAL.
- No span authorization checking is performed for an operator with a CTL=GLOBAL specification. CTL=GLOBAL is specified in the operator profile in DSIPRF or in the NETVIEW segment of the SAF product.
- The default value depends on the setting of SECOPTS.OPERSEC. When SECOPTS.OPERSEC=SAFDEF, SECOPTS.OPSPAN defaults to SAF. For other values of SECOPTS.OPERSEC, SECOPTS.OPSPAN defaults to NETV. For more information, see Table 13 on page 177.
- You can issue the REFRESH command with the OPSPAN keyword to dynamically change OPSPAN as long as it is compatible with the SECOPTS.OPERSEC specification.

SECOPTS.SPANAUTH

The SECOPTS.SPANAUTH statement specifies the location of the span definitions for resources and views. This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.SPANAUTH statement is:

```plaintext
SECOPTS.SPANAUTH={*NONE* | TABLE.span_table | VTAMLST.span_table}
```

Where:

*NONE*

- Specifies that no span definitions are defined or checked.

TABLE.span_table

- Specifies that the NetView program is to verify authorization for resources and views using the NetView span_table. The table can be modified and reloaded using the REFRESH command without requiring NetView to be recycled. For information on the NetView span table syntax, see the Tivoli NetView for z/OS Security Reference.

VTAMLST.span_table

- The NetView program supports the VTAMLST option in migration mode only. VTAMLST and DSISPN definitions are converted into a span table using the SECURMIGR command. The converted table is written to the first DSIPARM data set and is put into effect. Make sure that the PPT can write the table to this data set in your environment.

Usage Note: The SECOPTS.SPANAUTH setting is ignored when SECOPTS.OPERSEC=MINIMAL. With other SECOPTS.OPERSEC settings, you can issue the REFRESH command to change the method used for span of control authorization.
The SECOPTS.SPANCHK statement specifies the operator ID that is used to define span checking. For specific information on how the SOURCEID and TARGETID are determined, see [Tivoli NetView for z/OS Security Reference](#). This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.SPANCHK statement is:

```plaintext
SECOPTS.SPANCHK= SOURCEID
```

Where:

**SOURCEID**
- Specifies to check the authority of the original issuer or the ID closest to the original issuer.
- Access failure messages display the source issuer of the command. SOURCEID is the default.

**TARGETID**
- Specifies to check the authority of the task under which the command runs.

---

The SECOPTS.SURROGAT statement specifies whether the NetView operator ID is checked to determine if it is a surrogate of a TSO user ID. This statement is coded in CNMSTYLE.

The syntax for the SECOPTS.SURROGAT statement is:

```plaintext
SECOPTS.SURROGAT= NO
```

Where:

**NO**
- Surrogate checking is not performed.

**YES**
- Surrogate checking is performed if SECOPTS.OPERSEC is set to SAFCHECK in CNMSTYLE or OPERSEC=SAFCHECK is specified on the REFRESH command. This check is performed during START TSOSERV, STOP TSOSERV, and TSO PIPE stage processing.

---

The SECOPTS.WEBAUTH statement specifies whether authorization checking is to be performed for operator access to the NetView Web server. When checking is in effect, command authorization checking is performed using the WEBCMD command. Subsequent command authorization checking is performed against any commands entered by the operator. This statement is coded in CNMSTYLE.
SECOPTS.WEBAUTH

The syntax for the SECOPTS.WEBAUTH statement is:

```
SECOPTS.WEBAUTH= CHECK
PASS
```

Where:

**CHECK**

Perform authorization checking for access to the NetView Web server.

**PASS**

Do not perform authorization checking for access to the NetView Web server.

For performance reasons, if all NetView operators are to be granted access to the NetView Web server, specify a value of PASS. This is the default.

SENDMSG

The SENDMSG statement enables you to obtain additional status information from the status monitor.

Message CNM094I provides status information about resources defined to the status monitor. If DSIPARM member DSICNM contains an O SENDMSG statement, this message can be issued during status monitor initialization when VTAM notifies the status monitor that a resource known to it has changed status. When a resource passes through several status changes in rapid succession, message CNM094I might not be issued for the intermediate statuses.

Use the SENDMSG statement to specify each type of resource for which additional status information is needed. You code this statement in the status monitor initialization member DSICNM.

The syntax for the SENDMSG statement is:

```
SENDMSG
```

Where:

**resource_type**

Indicates the resource from which additional status information is needed. The SENDMSG statement must start in column 1 with `resource_type` starting in column 9. Code a SENDMSG statement for each resource type for which additional status information is needed.

If you request additional information for a type of resource and your network contains a large number of such resources, the status monitor then issues a large number of corresponding CNM094I messages. Use message CNM094I in conjunction with automation to enhance the recovery of resources in the network. Use the automation table entry for CNM094I to suppress the display and logging of this message as described in the automation definitions sample DSITBL01.

**Related Statements:** O SENDMSG
SOCKETS

The SOCKETS statement specifies how many sessions can be active at one time in the given NetView component using TCP/IP. TCP/IP reserves a minimum of 50 sockets, so numbers less than 50 are not used.

Note: The SOCKETS statement is used in the following samples:
- DSIILGCF
- DSIREXCF
- DSIRSHCF
- DSIRTTTD
- DSITCPCF
- DSIWBMEM
- DUIFPMEM

The syntax for the SOCKETS statement is:

```
SOCKETS
```

Where:

- `number`
  Specifies the number of sockets. Valid socket numbers are in the range of 50–2000. The default is 50.

Usage Notes:
- For the NetView Web server, a single blank must precede and follow the equal sign in the SOCKETS statement.
- The maximum value for the NetView 3270 management console is limited to 1000.

SPAN

Use the SPAN statement for span of control when SECOPTS.OPSPAN=NETV is specified in CNMSTYLE or OPSPAN=NETV is specified on the REFRESH command. SPAN is coded in an operator profile and defines the spans that an operator can add to span of control by using the START SPAN command. These spans are not in the operator’s control at logon.

You can use SPAN as often as necessary to define all the desired span names. You code this statement in the DSIPRF member specified by a PROFILEN statement associated with the operator. Changes made to the SPAN statement take effect the next time an operator logs on to the NetView program using the profile containing the statement. See "OPERATOR" on page 150 and "PROFILEN" on page 161 to determine how a PROFILEN statement is associated with an operator.

The syntax for the SPAN statement is:
SPAN

**Where:**

- **label**
  Indicates the optional label for the SPAN statement. This label identifies the statement in any related error messages.

- **spanname [,....]**
  Specifies the 1–8 character name of the span. You can use the same span name in more than one profile. These `spannames` are associated with resources through the NetView span table.

- **R**
  Specifies that an access level of READ is to be granted to `spanname`. This allows information-only access to resources and views defined to `spanname`. This level includes functions such as LIST and DISPLAY.

- **U**
  Specifies that an access level of UPDATE is to be granted to `spanname`. This allows change access to resources and views defined to `spanname`. This level includes functions such as VARY and the generic activate action.

- **C**
  Specifies that an access level of CONTROL is to be granted to `spanname`. This allows multi-read and single-write access to resources and views defined to `spanname`.

- **A**
  Specifies that an access level of ALTER is to be granted to `spanname`. This allows multiwrite access to resources and views defined to `spanname`. This is the default if no access level is specified.

**Related Statements:** AUTH, ISPAN, OPERATOR, PROFILEN

### SSLname

The SSLname statement specifies the task name and console name for the subsystem interface. Code this statement in CNMSTYLE.

The syntax for the SSLname statement is:

```
SSLname
```

**Where:**

- **name**
  Name of the CNMCSSIR task.

**Usage Notes:**
Comment out the SSIname statement if you do not need to use the CNMCSSIR task.

Within a sysplex, only one task is able to use the name CNMCSSIR (default task name). If there are other CNMCSSIR tasks running on other NetView programs within the same sysplex, use different task names to avoid name conflicts. For example, you could specify that SSIname has a value of C&NV2I.CSSIR to ensure that the value is unique for each CNMCSSIR task running within a sysplex.

STATOPT

Use the STATOPT operand values to control the functions of the status monitor. You code these operands following the applicable VTAM node definition statements. Code this statement in the member that defines a resource. To implement changes to rerun the status monitor preprocessor, stop and restart the task specified by TSKID xxxxxVMT. If changing the member results in additional extents being used, stop and restart the NetView program. After the preprocessor is finished, but before restarting the task, reaccess the minidisk containing DSINDEF.

Using the STATOPT operands, you can do the following:

- Code a specific description for the node in place of the default node description.
- Exclude the node from automatic reactivation by the NetView program.
- Exclude the node from status monitor activity displays.
- Omit the node, plus all the dependent lower nodes that follow, from the status monitor’s view of the network definition.

The basic types of STATOPT statements within major node definition statement members are as follows:

- A statement following an initial major node macroinstruction statement (LBUILD, VBUILD, or BUILD macroinstruction) that defines a descriptive name for the major node when the default is not desired.
- A statement following a VTAM or NCP macroinstruction definition statement for a minor node.

For an NCP major node that is to be monitored by the status monitor, verify that none of the following operands for the BUILD macro are continued from one line to the next: typgen, newname, puname. The status monitor preprocessor does not recognize the continued portion of an operand and unpredictable results can occur.

If the VTAMLST definitions specified for status monitor processing contain resources of the same name and type, only the first of these resource definitions processed by the status monitor is updated. For example, if an NCP major node, NCP001, contains a definition for LU1, and a backup major node, NCP002, also contains a definition for LU1, the status is monitored for the first LU1 found (in DSINDEF). The status monitor is unaware of the second LU1.

If NCP001 is activated first, then inactivated, and then NCP002 is activated, the status monitor might not display the correct status for either of the LU1 resources. If NCP002 is then inactivated and NCP001 is activated again, the status monitor displays the correct status for the LU1 under NCP001.

It is not necessary to code a STATOPT statement for every VTAM node definition. The defaults for a node that is not coded are given in the operand descriptions that follow.
The syntax for the STATOPT statement is:

```
STATOPT ( "nodedesc", "NOMONIT", "NOACTY", OMIT )
```

Where:

* Causes VTAM to treat the STATOPT statement as a comment statement. The asterisk must be in position 1.

**STATOPT=**

Is used to control major or minor node monitoring functions for the status monitor.

The rules for adding the STATOPT statement are:

- Insert the statements directly after the related VTAM macroinstruction definition. You cannot insert them between statements for the same VTAM macroinstruction (between continuation statements).
- Code each statement on a single record.
- If you use the network definition facility (NDF), specify your comment keywords on the GROUP macro after you have run NDF. If you do not, NDF generates PU and LINE statements for the GROUP macro and associates the comment keyword with only the last PU generated.
- STATOPT must start in position 16.
- All operands must follow directly after the equal sign (=). Do not leave a blank after the equal sign.
- The VTAM VBUILD statement must be present in application major node definitions or the STATOPT statements are not recognized by the preprocessor.
- If node desc is coded with one or more of the other operands, it must be the first operand. You can code the other operands in any order. Enclose node desc in single quotation marks.

**OMIT**

Excludes the node, plus all the dependent lower nodes, from the status monitor’s view of the network definition. If you do not code this operand, the node is included in the status monitor’s view of the network. When you specify OMIT after a VTAM definition for a cluster, the cluster and the terminals defined on the cluster are omitted from status monitor’s view of the network definition.

**nodedesc**

Indicates the 14-character description of any major or minor node. The description is displayed on the DESCRIPT form of the Status Detail panels.

The description must not include any single quotation marks. If you do not code a description, the STATOPT default descriptions for the node types in Table 15 on page 187 are used.
Table 15. STATOPT Default Descriptions

<table>
<thead>
<tr>
<th>Node Group</th>
<th>Default Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application major node</td>
<td>APPL MAJ NODE</td>
</tr>
<tr>
<td>Application minor node</td>
<td>APPLICATION</td>
</tr>
<tr>
<td>CDRM major node</td>
<td>CDRM MAJ NODE</td>
</tr>
<tr>
<td>CDRM minor node</td>
<td>CDRM</td>
</tr>
<tr>
<td>CDRSC major node</td>
<td>CDRSC MAJ NODE</td>
</tr>
<tr>
<td>CDRSC minor node</td>
<td>CDRSC</td>
</tr>
<tr>
<td>Local non-SNA terminal major node</td>
<td>LOCAL 3270 MAJ</td>
</tr>
<tr>
<td>Local non-SNA terminal minor node</td>
<td>LOCAL dev-addr</td>
</tr>
<tr>
<td>Local SNA PU/LU major node</td>
<td>LCL SNA MAJOR</td>
</tr>
<tr>
<td>Local SNA PU minor node</td>
<td>LOCAL SNA PU</td>
</tr>
<tr>
<td>Local SNA LU minor node</td>
<td>LOCAL SNA LU</td>
</tr>
<tr>
<td>NCP major node</td>
<td>NCP MAJOR NODE</td>
</tr>
<tr>
<td>Line minor node</td>
<td>LINE</td>
</tr>
<tr>
<td>Remote PU minor node</td>
<td>PU</td>
</tr>
<tr>
<td>Remote LU minor node</td>
<td>LU</td>
</tr>
<tr>
<td>Remote cluster minor node</td>
<td>CLUSTER</td>
</tr>
<tr>
<td>Remote terminal minor node</td>
<td>TERMINAL</td>
</tr>
<tr>
<td>Channel-channel major node</td>
<td>CA MAJOR NODE</td>
</tr>
<tr>
<td>Line minor node</td>
<td>LINE</td>
</tr>
<tr>
<td>PU minor node</td>
<td>PU</td>
</tr>
<tr>
<td>LU minor node</td>
<td>LU</td>
</tr>
<tr>
<td>CLUSTER minor node</td>
<td>CLUSTER</td>
</tr>
<tr>
<td>TERMINAL minor node</td>
<td>TERMINAL</td>
</tr>
<tr>
<td>Switched major node</td>
<td>SWITCHED MAJOR</td>
</tr>
<tr>
<td>Switched PU minor node</td>
<td>SWITCHED PU</td>
</tr>
<tr>
<td>Switched LU minor node</td>
<td>SWITCHED LU</td>
</tr>
<tr>
<td>LAN major node</td>
<td>LAN major node</td>
</tr>
<tr>
<td>Line minor node</td>
<td>LINE</td>
</tr>
<tr>
<td>PU minor node</td>
<td>PU</td>
</tr>
<tr>
<td>PACKET major node</td>
<td>PACKET major node</td>
</tr>
<tr>
<td>LINE minor node</td>
<td>Line</td>
</tr>
<tr>
<td>PU minor node</td>
<td>PU</td>
</tr>
<tr>
<td>LU minor node</td>
<td>LU</td>
</tr>
</tbody>
</table>

NOMONIT

Excludes the node from automatic reactivation. Code the NOMONIT operand on a STATOPT statement to prevent automatic reactivation. If this operand is not present, the node is included for automatic reactivation when monitoring is on. Do not code this option for major nodes or applications. See "O MONIT" on page 147 to determine how to enable monitoring.

NOACTY

Excludes the application node from activity recording. Code the NOACTY operand on a STATOPT statement to prevent activity displays. The default condition includes the node in the activity displays if the node is an application node.

Usage Note: If you code the STATOPT statement under the PCCU macro in the NCP major node, move the STATOPT statement after the BUILD macro.

Related Statements: O MONIT
styleMsg

The styleMsg statement specifies the message to be issued when CNMSTYLE processing begins. This statement is coded in CNMSTYLE.

The syntax for the styleMsg statement is:

```
styleMsg
```

Where:

`msg_text`

Specifies the message text to be displayed when CNMSTYLE processing begins.

Usage Notes:

- Consider using this statement to identify which CNMSTYLE member is being processed (display the value of &NV2I).
- You can specify as many styleMsg statements as necessary. The message text is displayed on the system console. After 70 characters are displayed, the message text for each statement is continued (wrapped) on the next line.
- Concatenation is supported. For each styleMsg statement, the message text can be continued on multiple lines. Enclose any variables in double quotation marks that you do not want resolved. See CNMSTYLE for coding examples.
- Variables are resolved in the message text, even when they are defined in statements that follow this statement in CNMSTYLE.

SUBSYSTEM

The SUBSYSTEM statement defines the DB2 subsystem to which NetView will connect. This statement is contained in member DSIDB2DF and is used by the DSIDB2MT task.

The syntax for the SUBSYSTEM statement is:

```
SUBSYSTEM
```

Where:

`subsystemname`

Name of the DB2 subsystem to which NetView connects. Subsystemname is a character value with a length of 1 to 4 characters.
SuppChar

The SuppChar statement specifies the suppression character to be used to prevent a command from appearing on the terminal screen, hardcopy log, or NetView log. This statement is coded in CNMSTYLE.

The syntax for the SuppChar statement is:

Where:

**SuppChar**


Where:

- **character**
  - Is the single character used as the suppression character. The question mark (?) is the default.

Usage Note: To prevent operators from suppressing command logging, comment out the SuppChar statement. The REXX SUPPCHAR() function will continue to work in REXX procedures.

TAMEL

The TAMEL statement defines TCP/IP definitions for CNMTAMEL (member DUIFPMEM). Code this statement in CNMSTYLE.

The syntax for the TAMEL statement is:

**TAMEL**


Where:

- **TCPANAME** = *name*
  - Defines the TCP/IP address space name.

- **PORT** = *port*
  - Defines the port number on which the workstation server waits for connection requests. The default is 4020.

- **SOCKETS** = *number*
  - Specifies the maximum number of simultaneous NETCONV sessions. The default is 50.

Usage Note: To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE TAMEL command.
The TASK statements in samples CNMSTYLE and CNMSTASK define a task to the NetView program. A task can be defined by one to four task statements as described below. Note that task statements can be split between multiple files: CNMSTYLE contains the task statements that are most commonly modified; CNMSTASK contains the task statements that should not be modified. For example, for task DSILOG, CNMSTYLE contains the INIT statement and CNMSTASK contains the MOD, MEM, and PRI statements.

You can define optional tasks (OPTs) and data service tasks (DSTs) in CNMSTYLE. If you do not want to define OPTs and DSTs in CNMSTYLE, use the START TASK command to dynamically start an OPT or DST. Refer to the online help for the START command for the syntax of the task operand.

The format of the TASK statement is:

```
TASK
  taskname.

  MOD= modulename

  MEM= membername

  PRI= n

  INIT= N | Y
```

Where:

**MOD=modulename**
Indicates a 1–8 character name of the module that runs a task. This is a required operand. For NetView data services tasks (DSTs), the module name is DSIZDST.

**MEM=membername**
Indicates the user-defined initialization member name that is used by this task. This is an optional operand. This task is responsible for the format and contents of this member. For IBM-supplied NetView program tasks, the initialization member is processed only when MOD=DSIZDST. For more information, see “DSTINIT” on page 41.

**PRI=9|n**
Indicates a number that defines the dispatching priority of this task in relation to other subtasks running in this NetView program. This is an optional operand. The value of n can be from of 1 to 9. The value of 1 is the highest priority you can assign, and 9 is the lowest. If you do not specify a priority value or you specify an incorrect value, 9 is the default. The primary POI task is priority 0, the hardcopy task is priority 2, autotasks are priority 5, the other operator station tasks and NetView-NetView tasks are priority 4.

**INIT=N | Y**
Specifies when the task begins:

- **N** Specifies that the NetView program START command must be issued to start the task. This is an optional operand. N is the default value.
- **Y** Specifies that NetView program initialization starts the task.

**Related Statements:** DSTINIT
The TASK statement enables you to define a NetView autotask or operator task that the NetView program-to-program interface command receiver uses to execute commands. You can define as many as 64 tasks to the program-to-program interface command receiver by coding multiple TASK statements in DSIQTSKI.

The syntax for the TASK statement is:

```
TASK
```

Where:

```
taskname
```

Specifies a valid NetView task name that can be 1–8 characters.

The TCPANAME statement enables you to specify the name of the TCP/IP address space.

Note: The TCPANAME statement is used in the following samples:

- DSIILGCF
- DSIREXCF
- DSIRSHCF
- DSIRTTTD
- DSIT CPCF
- DSIWBMEM
- DUIFPMEM
- DUIIGHB

The syntax for the TCPANAME statement is:

```
TCPANAME
```

Where:

```
proc
```

Specifies the name of the procedure used to start the TCP/IP address space. This keyword is required for use of the TCP/IP function.

Usage Note:

For the NetView Web server, a single blank must precede and follow the equal sign in the TCPANAME statement.
TCPname

The TCPname statement specifies the TCP name as a global variable. This statement is coded in CNMSTYLE.

The syntax for the TCPname statement is:

```
TCPname TCP_name
```

Where:

TCP_name Indicates a 1–5 character TCP name.

Usage Notes:

- If you set the system symbolic &CNMTCNP in SYS1.PARMLIB member IEASYMxx, do not update this statement.

TCPserver

The TCPserver statement defines the TCP/IP server to use for your commands. This statement is coded in CNMSTYLE.

The syntax for the TCPserver statement is:

```
TCPserver server_name
```

Where:

server_name Specifies a specific TSO server as defined by the TSOSERV definition statement in sample FKXCFG01.

TSO Specifies to use the next available TSO server defined by the TSOSERV definition statement in sample FKXCFG01.

UNIX Specifies to use the UNIX® server. This is the default.

TITLE

The TITLE statement enables you to change the color of the title line that appears on the command facility panel. The sample member containing the TITLE statement is CNMSCNFT. You can code the TITLE statement only once.

The syntax for the TITLE statement is:
TITLE

Where:

colorf
Defines foreground color for the command facility title line. The foreground color must be specified before the background color.

attribute
Defines alarm, intensity, and highlight attributes for the command facility title line. You can specify attributes only once for the TITLE statement.

ON
Makes the color following ON apply to the background of the title line. This is a required keyword if you specify only a background color.

colorb
Defines background color for the command facility title line.

Usage Notes:
• To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
• The NetView-supplied defaults for the TITLE statement are:
  – White foreground
  – Black background
  – Normal highlighting
  – High intensity
  – Alarm off
• The color operands are:
  BLACK   The color black
  BLUE    The color blue
  RED     The color red
  PINK    The color pink
  GREEN   The color green
  TURQ    The color turquoise
  YELLOW  The color yellow
  WHITE   The color white
• The highlight operands are:
  BLINK    The characters blink.
  REV      The characters change to reverse video.
  UND      The characters are underlined.
  NRM      Normal attributes are used.
• The intensity operands are:
  HIGH     The characters have high intensity.
  LOW      The characters have low intensity.
• The alarm operand is:
  BEEP     An audible alarm sounds.
The TILEDATE statement enables you to change the color of the date on the title line that appears on the command facility panel. The sample member containing the TILEDATE statement is CNMSCNFT. Code the TILEDATE statement only once.

The syntax for the TILEDATE statement is:

```
TILEDATE
```

**Where:**

colorf
  Defines foreground color for the date. You must specify the foreground color before the background color.

attribute
  Defines alarm, intensity, and highlight attributes for the date. You can specify attributes only once for the TILEDATE statement.

ON
  Makes the color following ON apply to the background of the date. This is a required keyword if you specify only a background color.

colorb
  Defines background color for the date.

**Usage Notes:**

- To create a member containing screen format definitions, use this member. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the TILEDATE statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - BLACK The color black
  - BLUE The color blue
  - RED The color red
  - PINK The color pink
  - GREEN The color green
  - TURQ The color turquoise
  - YELLOW The color yellow
  - WHITE The color white
The highlight operands are:

- **BLINK** The characters blink.
- **REV** The characters change to reverse video.
- **UND** The characters are underlined.
- **NRM** Normal attributes are used.

The intensity operands are:

- **HIGH** The characters have high intensity.
- **LOW** The characters have low intensity.

The alarm operand is:

- **BEEP** An audible alarm sounds.

**Related Statements:** ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDEXT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDOMID, TITLEOID, TITLESTAT, TITLETIME

---

**TITLEDOMID**

The TITLEDOMID statement enables you to change the color of the NetView program domain name that appears on the command facility panel. The sample member containing the TITLEDOMID statement is CNMSCNFT. You can code the TITLEDOMID statement only once.

The syntax for the TITLEDOMID statement is:

```
TITLEDOMID
```

Where:

- **color** Defines foreground color for the NetView program domain name. You must specify the foreground color before the background color.

- **attribute** Defines alarm, intensity, and highlight attributes for the NetView program domain name. Specify attributes only once for the TITLEDOMID statement.

- **ON** Makes the color following ON apply to the background of the NetView program domain name. This is a required keyword if you specify only a background color.

- **colorb** Defines background color for the NetView program domain name.

**Usage Notes:**

- You can create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.

- The NetView-supplied defaults for the TITLEDOMID statement are:
  - White foreground
  - Black background
The TITLEOPID statement enables you to change the color of the NetView operator identifier that appears on the command facility panel. The sample member containing the TITLEOPID statement is CNMSCNFT. You can code the TITLEOPID statement only once.

The syntax for the TITLEOPID statement is:

```
TITLEOPID (colorf) attribute (on colorb)
```

Where:

- `colorf` Defines foreground color for the NetView operator identifier. You must specify the foreground color before the background color.

- `attribute` Defines alarm, intensity, and highlight attributes for the NetView operator identifier. You can specify highlight and intensity attributes only once for the TITLEOPID statement.
ON
  Makes the color following ON apply to the background of the NetView
  operator identifier. This is a required keyword if you specify only a
  background color.

colorb
  Defines background color for the NetView operator identifier.

Usage Notes:
  • You can create a member containing screen format definitions. To use this
    member, specify the name of the member on the SCRNFMT parameter of the
    DEFAULTS or OVERRIDE command.
  • The NetView-supplied defaults for the TITLEOPID statement are:
    – White foreground
    – Black background
    – Normal highlighting
    – High intensity
    – Alarm off
  • The color operands are:
    BLACK   The color black
    BLUE    The color blue
    RED     The color red
    PINK    The color pink
    GREEN   The color green
    TURQ    The color turquoise
    YELLOW  The color yellow
    WHITE   The color white
  • The highlight operands are:
    BLINK   The characters blink.
    REV     The characters change to reverse video.
    UND     The characters are underlined.
    NRM     Normal attributes are used.
  • The intensity operands are:
    HIGH    The characters have high intensity.
    LOW     The characters have low intensity.
  • The alarm operand is:
    BEEP    An audible alarm sounds.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT,
HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT,
NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE,
TITLEDOMID, TITLETEM, TITLESTAT

TITLESTAT

The TITLESTAT statement enables you to change the color of the status indicators
at the end of the title line that appears on the command facility panel. The sample
member containing the TITLESTAT statement is CNMSCNFT. You can code the
TITLESTAT statement only once.

The syntax for the TITLESTAT statement is:
TITLESTAT

Where:

\textit{colorf}

Defines foreground color for the status indicators. You must specify the foreground color before the background color.

\textit{attribute}

Defines alarm, intensity, and highlight attributes for the status indicators. You can specify attributes only once for the TITLESTAT statement.

\textit{ON}

Makes the color following ON apply to the background of the status indicators. This is a required keyword if you specify only a background color.

\textit{colorb}

Defines background color for the status indicators.

Usage Notes:

- You can create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The following defaults are provided by NetView for the TITLESTAT statement:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - \texttt{BLACK} The color black
  - \texttt{BLUE} The color blue
  - \texttt{RED} The color red
  - \texttt{PINK} The color pink
  - \texttt{GREEN} The color green
  - \texttt{TURQ} The color turquoise
  - \texttt{YELLOW} The color yellow
  - \texttt{WHITE} The color white
- The highlight operands are:
  - \texttt{BLINK} The characters blink.
  - \texttt{REV} The characters change to reverse video.
  - \texttt{UND} The characters are underlined.
  - \texttt{NRM} Normal attributes are used.
- The intensity operands are:
  - \texttt{HIGH} The characters have high intensity.
  - \texttt{LOW} The characters have low intensity.
- The alarm operand is:
  - \texttt{BEEP} An audible alarm sounds.
Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLETIME

**TITLETIME**

The TITLETIME statement enables you to change the color of the time on the title line that appears on the command facility panel. The sample member containing the TITLETIME statement is CNMSCNFT. You can code the TITLETIME statement only once.

The syntax for the TITLETIME statement is:

```
TITLETIME
```

Where:

colorf

Defines foreground color for the time. You must specify the foreground color before the background color.

attribute

Defines alarm, intensity, and highlight attributes for the time. You can specify attributes only once for the TITLETIME statement.

ON

Makes the color following ON apply to the background of the time. This is a required keyword if you specify only a background color.

colorb

Defines background color for the time.

Usage Notes:

- You can create a member containing screen format definitions. To use this member, specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView-supplied defaults for the TITLETIME statement are:
  - White foreground
  - Black background
  - Normal highlighting
  - High intensity
  - Alarm off
- The color operands are:
  - **BLACK** The color black
  - **BLUE** The color blue
  - **RED** The color red
  - **PINK** The color pink
  - **GREEN** The color green
  - **TURQ** The color turquoise
  - **YELLOW** The color yellow
  - **WHITE** The color white
TITLETIME

- The highlight operands are:
  - BLINK: The characters blink.
  - REV: The characters change to reverse video.
  - UND: The characters are underlined.
  - NRM: Normal attributes are used.
- The intensity operands are:
  - HIGH: The characters have high intensity.
  - LOW: The characters have low intensity.
- The alarm operand is:
  - BEEP: An audible alarm sounds.

Related Statements: ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT

TOWER

The TOWER statement activates optional NetView components and other products. This statement is coded in CNMSTYLE.

The syntax for the TOWER statement is:

```
TOWER component_name
```

Where:

component_name

Identifies the tower name. Tower names are alphanumeric (includes @, #, and $) and are from one to sixteen characters.

Usage Notes:
- To disable a tower, either remove the name or prefix the name with an asterisk. For example, you enable the installation of MultiSystem Manager (MSM) by including MSM in the TOWER statement: TOWER = *AON MSM. To disable installation of MultiSystem Manager, remove MSM from the tower statement or prefix it with an asterisk (*MSM).
- Towers can have subtowers. For example, the MSM tower has a subtower for each of its features. For more information about the TOWER statement and subtowers, refer to sample CNMSTYLE and to the Tivoli NetView for z/OS Installation: Getting Started.

TRACE

The TRACE statement initializes the NetView tracing options. Code this statement in CNMSTYLE.

The syntax for the TRACE statement is:
TRACE

Where:

OPTION = DISP | MOD | PSS | QUE | STOR | TCP | UEXIT

Indicates which options are to be traced. Each option identifies an internal event type that is to be traced. The options are as follows:

DISP  Indicates dispatching of tasks including waiting (DSIWAT), post (DSIPOS), and dispatch from a wait (resumption of processing from DSIWAT).

MOD  Indicates module entry and exit trace of a subset of NetView modules.

Note: Using MOD severely degrades the performance of the system, therefore use MOD only to trap specified data.

PSS  Indicates presentation services, which involves input from and output to the terminal screen using DSIPSS.

QUE  Indicates inter-task queueing of buffers using DSIMQS.

STOR  Indicates getting and freeing of storage.

TCP  Indicates IP services related calls.

UEXIT  Indicates installation exit calls for:
- DSIEX01 through DSIEX19
- Resource status manager exit (XITST)
- CNM interface input exit (XITCI)
- CNM interface output exit (XITCO)
- DST initialization exit (XITDI)
- VSAM empty file exit (XITVN)
- VSAM input exit (XITVI)
- VSAM output exit (XITVO)

MODE = EXT | GTF | INT | *NONE*

Specifies in which area data is to be logged.
**TRACE**

| **EXT** | Indicates to log the trace data on the trace log. If you specify MODE=EXT, also specify INIT=Y for the trace task. |
| **GTF** | Indicates to log the trace data to the generalized trace facility (GTF). MODE=GTF is rejected if GTF is not active. |
| **INT** | Indicates to log the trace data in the internal table. INT is the default. |
| **NONE** | Indicates that tracing is off. |
| **SAFrc = calls** | Indicates calls made to an SAF product. Valid values for calls are: |
| | • ALL |
| | • FAIL |
| | • NONE |
| **SAFopt = options** | Indicates the types of SAF calls to trace. Valid values for options are as follows: |
| | • AUTH |
| | • EXTRACT |
| | • FASTAUTH |
| | • LIST |
| | • STAT |
| | • TOKENMAP |
| | • TOKENXTR |
| | • VERIFY |
| **SIZE = pages** | Indicates the number of pages of storage to allocate for in-storage trace table. If MODE=INT, the default page size is 999 pages (page size is 4k). If SIZE is specified with MODE=EXT or MODE=GTF, SIZE is ignored. |
| **TASK = task_type** | Specifies a task name or a task type. Valid values for task_type are: |
| | • ALL |
| | • HCT |
| | • MNT |
| | • NNT |
| | • OPT |
| | • OST |
| | • PPT |
| | • VOST |

**Usage Notes:**

- By default, TRACE is on, internal, with size 999, and options DISP, PSS, QUE, STOR, and UEXIT.
- To run without trace, specify TRACE.MODE=*NONE*.
- For more information, refer to the online help for the TRACE command.

---

**transMember**

The transMember statement defines the DSIMSG member that contains the national language translations for messages. For additional information, refer to the online help for the TRANSMSG command. Code this statement in CNMSTYLE. Stop and restart the NetView program to implement the changes.
transMember

The syntax for the transMember statement is:

```plaintext
transMember
transMember=membername

Where:
membername
DSIMSG member that contains the message translations.
```

transTbl

The transTbl statement defines a character translation set to the NetView program.
Code this statement in CNMSTYLE. Stop and restart the NetView program to
implement the changes.

The syntax for the transTbl statement is:

```plaintext
transTbl
transTbl=DSIEBCDC

Where:
DSIEBCDC
Selects an 8-bit coded character set called EBCDIC. This is the default.
DSIKANJI
Selects a character set of symbols used in Japanese ideographic alphabets
called kanji.
```

Usage Notes:
- All devices must use the same character set for meaningful results.
- The transTbl value for the log printer program should be the same value as the
  value used for the NetView program definition. The NetView program does not
  check these values for compatibility.
- If you define more than one transTbl statement, the NetView program uses the
  last one you entered.

VALCLASS

The VALCLASS statement is obsolete. It is used for migration purposes only. For
more information, refer to the Tivoli NetView for z/OS Installation: Migration Guide

Related Statements: CMDCLASS, KEYCLASS, OPCLASS
The VBV statement initializes the Visual BLDVIEWS server. Code this statement in CNMSTYLE.

The syntax for the VBV statement is:

```
VBV

CheckDDReadRODM = No | Yes
CheckDDWriteRODM = No | Yes
MaxRODMViewLines = lines
Port = port
Quiet = No | Yes
RODMUser = userid
trace_option = No | Yes
```

Where:

CheckDDReadRODM = No | Yes

Specifies whether to verify the RODM sign-on authorization sent by the VBV client before permitting the reading of any host data set. The default is No.

CheckDDWriteRODM = No | Yes

Specifies whether to verify the RODM sign-on authorization sent by the VBV client before permitting writing to any host data set. The default is No.

MaxRODMViewLines = lines

Specifies the maximum number of lines returned from RODM. If the value specified at the VBV client workstation is smaller than the one specified in CNMSTYLE, the workstation value will be used. For values greater than 0 and less than 100, the number of lines is set to 100. Zero is the default.

Port = port

Specifies the TCP/IP port to which VBV workstation clients connect. The default is 6767.

Quiet = No | Yes

Specifies whether the EKGBSERV task issues a message when it starts and finishes serving a request.

RODMUser = userid

Specifies the RODM user ID that is used by VBV to access RODM. This value should be set to an SAF-authorized user ID with the authority to sign on to RODM and to query RODM data. This name should be different from the RODMView user name that a workstation user would specify.

trace_option = No | Yes

Specifies whether to trace VBV events as directed by Tivoli Customer Support. The default is No. Using the trace option might generate large amounts of output to the NetView log. The trace_option can be one of the following:

- Trace
- TraceData
- Trace RODM
- TraceSockets
**Usage Note:** To implement definition changes while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE VBV command.

---

**VPDINIT**

The VPDINIT statement defines the operands supplied to subtask VPDTASK at initialization. VPDTASK is the NetView program subtask that collects and logs vital product data (VPD). Code this statement in the member specified by the MEM keyword on the VPDTASK TASK statement where MOD=DSIVMATK. The sample member supplied with the NetView program is DSIVPARM. To implement changes to the VPDINIT statements, stop and restart the VPDTASK. If changing the member results in additional extents being used, stop and restart the NetView program before the changes take effect.

The syntax for the VPDINIT statement is:

```
VPDINIT

VPDINIT ACBNAME=acbname, PASSWORD=password, VPDWAIT=nnn

VPDREQ=nnn, SNAPRQ=OFF, ON

VPDSTOR=nn
```

*Where:*

**ACBNAME=acbname**

Specifies the 1–8 character ACB name defined for subtask VPDTASK on the VTAM APPL statement. The NetView program uses ACBNAME to open its interface with VTAM. ACBNAME must match the ACBNAME specified in VTAMLST.

**PASSWORD=password**

Specifies the 1–8 character password associated with ACBNAME. If you code a password on the VTAM APPL statement in VTAMLST for subtask VPDTASK, code the same password here.

**VPDWAIT=30|nnn**

Specifies how long the NetView program should wait for a response to a request for VPD before timing out. Depending on the relative time between the last timeout check and the VPD request submitted, the NetView program waits for a response to the request between nnn seconds and two times nnn seconds before timing out. The value of VPDWAIT must be a number from 30–999 seconds. The default is 30.

**VPDREQ=1|nnn**

Specifies the maximum number of queued requests for VPD that VPDTASK accepts for processing. The request for VPD can be VPDCMD OWN, VPDCMD ALL, or VPDCMD DCE. The value of VPDREQ must be a number from 1–999. The default is 1.
VPDINIT

SNAPRQ=OFF|ON
Specifies whether the SNAP trace option should be turned on at VPDTASK initialization. The SNAP option can be turned on or off later, using the VPDCMD SNAP ON (or OFF) command. The default is OFF.

VPDSTOR=2|nn
Specifies the number of kilobytes allocated for formatting VPD that comes back in response to a VPD request. The value of VPDSTOR must be a number from 2–32. The default is 2.

There is no standard way to determine the optimal value for VPDSTOR. If you receive message DWO019 (insufficient storage), increase the value until you no longer receive the message.

VTAMCP

The VTAMCP statement specifies whether the NetView SNA MS transport running under a specific NetView program is able to receive MDS-MUs with the VTAM control point (CP) name as the destination. Use the VTAMCP statement when you have multiple NetView programs running under a single VTAM program.

Code this statement in CNMSTYLE. Changes to the VTAMCP statement do not take effect until you stop and restart the NetView program.

VTAMCP

Where:
NO
Indicates that this NetView program is not designated to receive MDS-MUs with the CP name as the destination.

YES
Indicates that this NetView program is designated to receive MDS-MUs with the CP name as the destination. This is the default.

Usage Notes:
- The first NetView program to specify VTAMCP.USE = Yes can receive MDS-MUs with the VTAM CP name as the destination, provided the program-to-program interface (PPI) is active. The remaining NetView programs use the management services transport function. As a result, the remaining NetView programs receive only MDS-MUs with the NetView LU name as the destination. For more information, refer to the Tivoli NetView for z/OS Application Programmer’s Guide.
- Only the first NetView program to initialize this function enables it. Subsequent attempts to initialize this function by other NetView programs fail.
- You do not need to change existing NetView command processors that issue MDS send requests to have them work with the Version 2 Release 4 or later of the NetView program. The MS transport function operates at the Version 2 Release 2 or later release level of the NetView program.
- The program-to-program interface must be active on the NetView program for which VTAMCP.USE = Yes is defined.
• IBM recommends that you use the same value for VTAMCP USE as you specified for the CNMI statement, unless instructed otherwise.

WEB

The WEB statement defines TCP/IP definitions for NetView Web access. Code this statement in CNMSTYLE.

The syntax for the WEB statement is:

WEB

Where:

TCPANAME = name
Defines the TCP/IP address space name. This is used by the Web access interface task (DSIWBTSK).

PORT = port
Defines the port number on which the Web access interface task (DSIWBTSK) waits for connection requests. The default is 8008.

SOCKETS = number
Specifies the maximum number of simultaneous Web browser sessions. This is used by the Web access interface task (DSIWBTSK). The default is 50.

NMC.function = url
Specifies the Web address for one of the following NetView management console functions:

function url

MIBBrowserURL
MIB browser

RealTimPollURL
Real time poller

SNMPCommandURL
SNMP and remote ping commands

Usage Note: To implement definition changes for NetView Web access while the NetView program is active, change the value in CNMSTYLE, and enter the RESTYLE WEB command.

WLM

The WLM statement activates NetView MVS workload management (WLM) support. This statement is coded in the CNMSTYLE file.

The syntax for the WLM statement is:
WLM

Where:

name

Specifies the SubSystemName value. This name should correspond to the system instance name specified in the WLM service classification rules.

Usage Notes:

• For more information, refer to sample CNMSTYLE.
This chapter applies to the AON automation component of Tivoli NetView for z/OS. It describes the policy definitions that you customize for:

- Base AON functions
- SNA and TCP automation components
- Inform policy for beeper and e-mail requests

### Defining AON Control File Entries

The AON control file entries determine how AON-based automation responds to events, such as resource failures, in your network. The AON control file supports system symbolics and is stored for performance purposes. The control file entries specify:

- Resource monitoring
- Environment specifications
- Resource threshold values
- Automation modes
- Message forwarding
- Automation operators
- Notification operators
- Notification policy

This chapter is a reference for system programmers coding the automation policy in the control file. The types of control file entries are:

- Automation entries that describe the automation environment
- Processing entries that define special handling of a member during load processing (%INCLUDE)

The entries are listed in alphabetical order. Entry descriptions include format, parameters, and where applicable, usage notes and examples.

#### Restriction about updating the control file

If the control file entry is more than 200 characters in length, you cannot update the entries online. If you need to make an entry that is longer than 200 characters, the update must be made offline.

### ACTMON

The ACTMON control file entry defines active monitoring policy for critical resources in the network. When a monitored resource fails, AON sets a timer to check the status of the resource. If the resource is not active, AON initiates failure processing and suspends active monitoring in favor of recovery monitoring.

#### ACTMON

```
ACTMON DEFAULTS, INTVL=timedintvl
```

#### Parameters

- **option**
- **restype**
- **resname**
ACTMON

\[-,\text{STATUS}=\{\text{status, status1, status2, statusn}\}\rightarrow, \text{OPTION}=\text{option}, \text{RESTYPE}=\text{restype}\rightarrow\]

\[-,\text{OPID}=\text{autoop_cglobal}, \text{SP}=\text{NV6000_servicepoint}, \text{PINGRTT}=\text{pingrtt_millisecs}\rightarrow\]

\[-,\text{PKTLOSS}=\text{loss }\%ightarrow\]

Where:

**DEFAULTS** | option | restype | resname

**DEFAULTS**
Required if you are going to perform any active monitoring. This statement defines any AON (any component) active monitoring definitions. Valid parameters for this statement are INTVL, STATUS, and OPID.

**option**
Required. For each option installed with AON (as defined by your INSTALLOPT control file statements) you may have an ACTMON definition containing active monitoring definitions that apply to any resource managed by that option. This statement does not initiate any active monitoring actions or timers. If you do not want to define any option-wide defaults, this statement is not necessary. The OPTION keyword must be defined and must be the same as the type field of this entry. RESTYPE is not a valid keyword for this entry.

The option specified on the ACTMON statement for a resource must match the option definition for the resource in the loader table. For example, to define ACTMON for a session, its option parameter must match the EZLRT definition for SESSION in member FKVTABLE. Specify OPTION=SA on the ACTMON statement for the session.

**restype**
Not required. This statement defines active monitoring definitions that apply to all resources of a particular type. These must be AON or component resource types as indicated in the respective component’s Automation Guide and the definition tables. The OPTION and RESTYPE keywords are required on this statement. The value of the RESTYPE keyword must be the same as the restype in the type field of this entry.

**resname**
Required for active monitoring. This statement causes active monitoring timers to be set for the resource. The OPTION and RESTYPE keywords are required for this statement. All the other keywords must be defined either at this level or at the restype, option, or DEFAULTS level ACTMON statements. The values on the most specific definition take precedence. For TCP/IP resources, resname can be either an IP address or a HOSTNAME.

**INTVL**
Required at some level. This value may be defined differently for each resource you want to monitor, or may be defined at any non-specific level definition (DEFAULTS, option, or restype). The format is \(hh:mm\) or \(mmm\). If ACTMON IPPORT is not defined, the default is 10 minutes.

**STATUS**
Required at some level. This value can be defined differently for each resource
to be monitored or at a nonspecific level definition (DEFAULTS, option, or restype). Valid values are those defined with the EZLSTS command. Parentheses are optional for a single status.

You can use a wildcard character in the last position and a negation symbol in the first position of the STATUS value. For example, if you specify STATUS=ACTIV, AON searches for an explicit status match of ACTIV. If you specify STATUS=ACT*, AON searches for a status match that begins with ACT. If you specify STATUS=/ACTIV, AON searches for all status except ACTIV. If you specify STATUS=/ACT*, AON searches for all status except those beginning with ACT.

**OPTION**
Required for every statement (except ACTMON DEFAULTS). This indicates the AON automation component that is responsible for automating the resource to be actively monitored. ACTMON definitions at this level can be applied to active monitoring against resource types that are supported by the particular automation component. Valid values for this field are the AON automation components defined in the INSTALLOPT option definition of the control file.

**RESTYPE**
Required for every statement (except ACTMON DEFAULTS and ACTMON option). This indicates the AON resource type to be actively monitored. Any ACTMON definitions at this level can be applied to active monitoring against resources of a particular type. Valid values for this field are any AON resource type definitions defined in the ETLRT statement of the AON option definition table (EZLTABLE, FKVTABLE, or FKXTABLE).

**OPID**
Optional. This value may be defined differently for each resource to be monitored, or may be defined at any non-specific level definition (DEFAULTS, option, or restype). Valid values are those defined as valid AUTOOPS in the control file AUTOOPS statements. If there are two operator IDs on a single AUTOOP statement, the first ID is referred to by the AUTOOP type field and the second is referred to by the same name with a 2 appended to it. If this keyword is not coded, the INSTALLOPT ACTMONOP definition is used. All active monitoring activities are routed to this operator ID. You can add AUTOOPS statements and automation operators to distribute the active monitoring workload.

**SP**
Required and is valid only for resources managed by AON/TCP. Defines which NetView for AIX® or TCP/IP for 390 service point reports and manages the resource and which service point AON routes status queries to (pings) for active monitoring. There must be a NV6000 or TCP390 definition that matches the value in this parameter. See "NV6000 (TCP/IP)" on page 263 for more information about NV6000 definitions or "TCP390" on page 274. You can define this value differently for each resource you want to monitor, or you can define it at any nonspecific level definition (DEFAULTS, NVAIX, TCP390 or resource_type).

**PINGRTT (NetView for UNIX)**
Optional and is valid only for resources managed by AON/TCP. Defines how long a round trip ping can take before a performance problem is recognized. This value is defined in milliseconds only. If the average round trip time exceeds this value during an active monitoring ping, AON/TCP notifies operators that the performance threshold has been exceeded. You can define this value differently for each resource you want to monitor, or you can define the value at any nonspecific level definition (DEFAULTS, NVAIX, or...
resource_type). In the next example, if the average ping round trip time exceeds
five milliseconds, AON/TCP sends a warning to operators, logs, and DDF:
ACTMON TCPGUY,OPTION=NVAIX,RESTYPE=HOST,PINGRTT=5

PKTLOSS (NetView for UNIX)
Optional and is valid only for resources managed by AON/TCP. Defines what
percentage of sent packets can be lost during the active monitoring ping before
a performance problem is recognized. You can define this value differently for
each resource you want to monitor, or you can define it at any nonspecific
level definition (DEFAULTS, NVAIX, or resource_type). This is defined in full
percentage points only. The % sign must be the last character. If the packet loss
percentage exceeds this value during an active monitoring ping, operators are
notified that the performance threshold has been exceeded. In the next example
if the packet loss for the ping to TCPGUY exceeds 50%, AON/TCP updates the
operators, DDF, and logs:
ACTMON TCPGUY,OPTION=NVAIX,RESTYPE=HOST,PKTLOSS=50%

Usage Notes:
1. Use a corresponding TCPIP control file entry for NVAIX resources to be
   actively monitored.
2. The active monitoring function checks the availability of a router’s IP address
   and the IP address of its links (interfaces). If AON/TCP cannot ping a link, it
   starts failure processing for that link. AON/TCP also puts the router in a
   LINKDOWN status, but active monitoring continues.
3. The active monitoring function causes both the host name and IP address of a
   name server to be pined. If the host name cannot be pinged, the name server
   is not operational and AON/TCP starts failure processing.

Example:

Following is an example of the ACTMON DEFAULTS control file entry that is
required if AON performs active monitoring:
ACTMON DEFAULTS,INTVL=01:00,STATUS=ACT

Active monitoring is started every hour for each ACTMON entry and the specific
resource that is defined to AON. If the status is ACT (also matches ACTIVE and
ACT/S), the resource is available and active monitoring is rescheduled. If the
status is not ACT, failure processing with the EZLEFAIL routine is started for the
resource. These values can be overridden at any other level (option, restype, or
resname).

Example:

The following is an example of active monitoring of AON/SNA subarea option
resources:
ACTMON SA,OPTION=SA,INTVL=00:30,OPID=NETOPER2

For every actively monitored resource managed by the AON/SNA subarea option,
the monitoring interval is every 30 minutes. The automation operator ID
NETOPER2 (defined in CGLOBAL) starts the active monitoring. These values can
be overridden at the restype or resname level. The STATUS value (in this case ACT)
is taken from the ACTMON DEFAULTS entry unless it is overridden at a lower
level.

Example:
In this example, any resources that are of an NCP resource type (PU T4/5) are actively monitored every 10 minutes. All other values, such as STATUS and OPID, must be defined at the ACTMON DEFAULTS or ACTMON SA level, or defined for each NCP to be actively monitored.

ACTMON NCP,OPTION=SA,RESTYPE=NCP,INTVL=00:10

Example:

In this example, active monitoring is started for resource NCP1, which is an NCP. Using the previous examples, it is monitored every 10 minutes (from the ACTMON NCP statement) on automation operator NETOPER2 (from the ACTMON SA statement) looking for a status of ACT* (from the ACTMON DEFAULTS statement).

ACTMON NCP1,OPTION=SA,RESTYPE=NCP

Example:

In this example, PU01 is a PU resource type and managed by the AON/SNA subarea option:

ACTMON PU01,OPTION=SA,RESTYPE=PU

Active monitoring is influenced by the following ACTMON definitions, if they exist:

ACTMON DEFAULTS,...
ACTMON SA,OPTION=SA,...
ACTMON PU,OPTION=PU,RESTYPE=PU,...

Example:

The PU resources managed by the AON/SNA subarea option are actively monitored every hour. NCP resources are actively monitored every 10 minutes. NCP1 is monitored every 30 minutes. NCP2 is monitored every 10 minutes (default).

ACTMON PU,OPTION=SA,RESTYPE=PU,INTVL=01:00
ACTMON NCP,OPTION=SA,RESTYPE=NCP,INTVL=10
ACTMON NCP1,OPTION=SA,RESTYPE=NCP,INTVL=30
ACTMON NCP2,OPTION=SA,RESTYPE=NCP

Example:

Any resources managed by the AON/TCP NetView for AIX option must have a status of NORMAL to pass the active monitoring cycle. If the status is not NORMAL, failure processing is initiated by calling the EZLEFAIL routine. This value may be overridden at the restype or resname level.

ACTMON NVAIX,OPTION=NVAIX,STATUS=NORMAL

Example:

In this example, AONNET2 starts active monitoring for resources managed by the AON/SNA subarea option, except for NCP active monitoring, which is started by AONBASE:

AUTOOPS NETOPER, ID=(AONNET,2),...
AUTOOPS BASEOPER, ID=AONBASE,....
INSTALLOPT SA, ACTMONOP=NETOPER2,....
ACTMON NCP, OPTION=SA, RESTYPE=NCP, OPID=BASEOPER
ADJNETV

The ADJNETV control file entry identifies domains used to route commands and replies to forward notifications between one NetView domain and another. The ADJNETV control file entry is optional.

```
ADJNETV dom1,DOMAIN=dom2,ALTNETV=dom3,DESC='description'
```

*dom1*  
Specifies the domain name to which the commands, replies, or notifications are being forwarded.

**DOMAIN**  
Identifies the NetView domain through which the commands, replies, or notifications can be forwarded.

**ALTNETV**  
The alternative or backup domain through which commands, replies, or notifications can be forwarded if the domain specified in the DOMAIN keyword is inactive.

**DESC**  
Specifies a descriptive term for this domain so that it can be easily identified.

**Usage Notes:**

You can view the status of adjacent NetView domains through the AON operator interface. The ADJNETV control file entry is not required if you can directly connect the target and focal point domains by an NNT session.

**Attention:** Domains that you specify as adjacent NetView programs must be physically adjacent to both the local system and the system to which the command or replies are being sent. Physically adjacent means that you can establish NNT sessions between the domains.

**Examples:**

The following ADJNETV entry assumes that CNM01 is the name of the current domain.

```
ADJNETV CNM03,DOMAIN=CNM02,ALTNETV=CNM99,DESC='PASSTHRU TO CNM03'
```

The ADJNETV entry in this example specifies that any commands, replies, or notifications that domain CNM01 forwards to domain CNM03 should pass through domain CNM021. However, if domain CNM02 is inactive, domain CNM99 is the alternate domain for passing the information to domain CNM03.

AUTOOPS

The AUTOOPS control file entries define NetView operator IDs that function as automation operators. Automation operators are AON tasks that respond to network messages without requiring operator intervention. Each automation operator initiates actions through procedures defined in the control file. Automation operators are assigned specific messages to act upon. Messages are assigned on the MSG parameter of the AUTOOPS entry.
AON defines multiple automation operators. Each operator processes as a separate
task within NetView. This task design permits NetView to distribute its workload
among the automation operators, thereby improving system performance.

AUTOOPS

Operid
Defines a name for the automation operator. Each automation operator relates
to a specific NetView operator ID by using the AUTOOPS ID parameter.

The AON command lists use the automation operator name to route
commands from one automation operator to another. During processing, the
associated NetView operator ID is substituted. This technique enables the
command list coding to be independent of the NetView operator IDs, which
may be unique in each installation. The value specified for the automation
operator name creates a NetView common global (CGLOBAL). The NetView
operator ID is stored in this common global (CGLOBAL).

Each AUTOOPS entry creates two common global variables; the first is the
automation operator name, the second is the automation operator name with a
2 appended at the end. Examples of automation operator names are MSGOPER
and MSGOPER2. If only one ID is defined, both CGLOBALs contain the same
name and ID.

The automation operator names provided with AON are:

AIPOPER
Sets and resets the AIP (Automation In Progress) operator status bit in
RODM. This bit causes a display pattern to be placed on the object in
NetView management console. RODM AIP operators issue the commands
necessary to update resource objects in RODM views with the AIP operator
status. These operators are also used in the management of the OIV
processing.

ALRTOPER
Sends alerts and resolutions to NetView over an LU 6.2 session.

BASEOPER
Provides backup for other automation operators.

DVIPOPER
Used for DVIPA polling processes.

GATOPER
The outbound gateway operator for automation notification forwarding.

INFOPER
Serializes the updates to the inform log.
**AUTOOPS**

**MSGOPER**
Formats and issues AON notifications and DDF updates.

**NETOPER**
Initiates routines based on the NetView automation table and AON generic failure and recovery routines.

**NV6KOP**
Used for Tivoli NetView for UNIX automation

**OIVOPER**
An optional operator task used by the Operation Intervention View (OIV) function. When enabled, automatically deletes resources from the OIV at specified intervals. Only resources with the display status of satisfactory (129) are removed.

**TCPOPER**
Used for TCP/IP automation

**TRAPOPER**
Used for trap automation processes

**WKSTOPER**
Sends and receives commands and responses between AON and a workstation with the interface installed.

**X25OPER**
Used by X25 automation processes

User-defined automation operator names can be added by defining a 1–10 character name, without embedded blanks, commas, single quotations, or periods, and cannot begin with a number.

**ID**
Defines a NetView operator ID used for an automation operator. Each operator ID must be defined to NetView in the DSIOPF member. The primary NetView operator ID receives all the incoming messages assigned to this automation operator. The secondary operator is defined for backup or off-load purposes only.

AON only supports off-loading work of NETOPER, MSGOPER, TCPOPER, and AIPOPER. You should provide at least two IDs for those automation operator definitions. Code a root TASKID to be used along with a number to identify how many autotasks are needed. For example, to define five AIPOPER autotasks, AUTAIP1 through AUTAIP5:

```
AUTOOPS AIPOPER,ID=(AUTAIP,5)
```

**Note:** When coding any AUTOOPS definitions, follow the syntax provided in the sample policy definitions shipped with AON.

The outbound gateway operator is defined as a NetView operator. The default naming convention for the operator ID is the 3-letter prefix GAT, followed by the *domain-ID*.

You can change the task names as needed for your installation.

**SEC**
Specifies one or more NetView operators to receive a copy of the messages defined for this automation operator. You can also specify a valid NetView group. All operators in this list receive the message if they are logged on and at least one operator in the ID list or one group ID is logged on.
This parameter also uses the NetView STATMON parameter GROUP. By defining a group name here, you can send a secondary message to all operators belonging to the group. You define groups in the NetView CLIST CNME1035 or in any other CLISTs that run early in NetView initialization. You can change the task names as needed for your installation.

**MSG**

Specifies the messages to be routed to this automation operator. AON uses the NetView ASSIGN statement to route these messages. You can use a prefix and an asterisk (*) to specify groups. For example, to specify all messages beginning with DSI, use DSI*. To specify all messages, use just an asterisk (*). Do not assign messages to BASEOPER and GATOPER parameters.

**Note:** SNA Automation requires specific VTAM messages to be assigned.

**Usage Notes:**

AON issues the NetView message assignment statements based on the order of the automation operators in the control file. The assignment statements must not overlap and be placed in order from most specific to least specific.

You must define at least one automation operator, referred to as the base operator (BASEOPER), with an AUTOOPS entry. Several automation operators are provided for optimum AON operation. However, you should also define additional automation operators when developing your own extensions to the automation process.

The primary operator assigned in the AUTOOPS ID statement is the first receiver of an assigned message. If the primary operator is logged off or abends, the secondary operator receives the message. If a secondary operator is not defined or is not logged on, the automation operator (BASEOPER) receives the message. This sequence is designed to provide a level of redundancy to ensure that automation continues if one or two automation operators were logged off or not functioning. The command list must be run to the secondary operator. Load balancing is not dynamic.

**Attention:** If you change the primary assignments provided with AON, the results may be unpredictable.

In situations where a command list runs for a long period of time, define a secondary operator. The secondary operator can process the command list, freeing up the primary operator to handle incoming messages.

When using Resource Access and Control Facility (RACF), it is not necessary to define the automated NetView operator IDs (host). The access checking of RACF is performed only when an operator (host) or another NetView system attempts a logon (NNT). The checking is bypassed if an automation operator is started through the AUTOTASK command. Therefore, when a NetView automation operator ID is not defined to RACF, an operator is not authorized to use that ID.

Gateway operators defined by GATOPER must be defined to RACF. GATOPERs on the focal point must be defined to RACF. If your installation is controlled by ACF2, define automation operators to ACF2, but not to RACF.

If an automation operator is not active, (NetView message DSI008), AON restarts the task. AON builds and initiates the NetView AUTOTASK command based on...
the DSI008 message. NetView issues tDSI008 when an automation operator is sent an element of work to perform, but it is determined by NetView that the operator is not active.

The following examples of AUTOOPS control file entries are SNA examples.

Example:

The following is an example of an automation operator assignment by the AUTOOPS entry:

```
AUTOOPS GATOPER, ID=GATCNM01
```

In this example, the name of the automation operator is GATOPER. The NetView operator ID, defined in the DSIOPF member, is GATCNM01. The ID is not enclosed in parentheses because only a single NetView operator ID is defined. GATOPER has no messages assigned to it. GATOPER is an outbound gateway operator that establishes and maintains the connections to other domains to enable automation notification forwarding.

Example:

The following are examples of additional automation operator definitions by AUTOOPS:

```
AUTOOPS BASEOPER, ID=AONBASE
AUTOOPS MSGOPER, ID=(AONMSG, 2), MSG=EZL*
```

Example:

In the following example, secondary automation operators are added using the AUTOOPS entry:

```
AUTOOPS NETOPER, ID=(AONNET, 5),
    SEC=(+OURGRP, OPER6),
    MSG=(CNM*, DSI*, EMS*, BNJ*)
```

In this example, the system programmer adds all operators to the group +OURGRP, which must be created by a user program that issues the ASSIGN command, and specifies that OPER6 get copies of all messages assigned to NETOPER. These copies are not subject to automation.

Example:

In the following example, nine automation operators are defined for AON messaging:

```
AUTOOPS MSGOPER, ID=(AONMSG, 9),
    MSG=(EZL*)
```

In this example, the system programmer increases the AON default of two MSGOPER autotasks to nine. The task names used will be AONMSG1 through AONMSG9. AON contains ten MSGOPER autotasks which are defined in EZLOPF. If the system programmer chooses to define 15 MSGOPER autotasks, EZLOPF must be modified to add the additional task definitions.

Example:

In the following the operator ID that is used for automatic deletion of resources from the Operator Intervention network View (OIV) is defined:
A timer statement in the control file is required to activate the automatic deletion.

**CDLOG**

The CDLOG control file entry defines which domains are displayed on the cross-domain logon (CDLOG) panel. Through the CDLOG component, you can log on to all or a select group of domains that are in your domain. The CDLOG interface lets you select which domains you want. The CDLOG control file entry is optional. If you do not define CDLOG control file entries all domains known to NetView are shown on the CDLOG panel. If you define CDLOG control file entries, only the domains defined are displayed on the CDLOG panel.

If you are running AON/TCP for z/OS across multiple domains, define RMTCMD sessions for each GATOPER with CDLOG definitions.

```plaintext
CDLOG operid.domainid,,SESSTYPE=NNT,TARGOP=operid2
```

- **operid**
  - ID of the NetView user initiating CDLOG.

- **domainid**
  - Target NetView domain to which session is to be established.

**SESSTYPE**

Identifies the type of session.

- **NNT** NetView-to-NetView task session (LU1).
- **RMT** NetView RMTCMD session (LU 6.2).

**TARGOP**

Target NetView operator to log on. Optional. If not specified, it defaults to the operator issuing the CDLOG command.

**TARGPW**

Password to use during session logon. The default is PROMPT.

- **PROMPT**
  - Prompt the operator for a password when establishing the session.

- **pswd**
  - The password specified in this variable is used.

**INIT**

Identifies the session to be started when the NetView operator (operid) logs on or when requested through the AON operator interface.

- **Y** Automatically logon the cross domain session when operid is logged on.
- **N** Let the operator choose which sessions to establish through the CDLOG command. NO is the default.

**DESC**

User text to identify the session.
CDLOG

Usage Notes:

Separate each parameter by a comma and follow all control file entry guidelines. To use CDLOG to log on cross-domains, remove all DSI809A statements from DSICMD. NetView operators who want to start NNT sessions without using CDLOG, receive the following message:

DSI809A PLEASE ROUTE OPID,PSWD,PROFILE,HARDCOPY,INITIAL CMD (optional: ,NEW PSWD, NEW PSWD)

Operators must route the appropriate information to continue the logon. RMTCMD sessions are not affected by DSI809A.

Before coding TARGPW, use a security product to protect operator passwords. Protect the EZLCFG, DSPCFG, and BROWSE commands. Code RRD statements in CNMSTYLE for cross-domain logon. It is not necessary to define RMTCMD sessions in CNMSTYLE.

Include CDLOG definitions for operators who are responsible for automating resources in domains other than their own. For example, the SNA Help Desk route activates commands over the operator’s CDLOG session.

Examples:

In the following example, a minimum CDLOG entry establishes a session between CNM01 and CNM1A for operator OPER1.

CDLOG OPER1.CNM1A,SESSTYPE=RMT

The CDLOG entry has the following default values:

INIT=NO

TARGOP

Logs on OPER1 in domain CNM1A.

TARGPW

Ignores password entry because this is a RMTCMD session.

DESC

Does not display user text on the CDLOG panel.

community_name (FKXSCM)

AON uses the community_name statement to control the access that the NetView program has to managed objects in the network. Community names are used in SNMP requests. The community name is similar to a password in that it determines whether an entity can gain access to information or perform an action. This statement is coded in FKVXCM.

The syntax for the community_name statement is:

community_name

community_name-ipaddr-hostname

Where:
community_name

Indicates the name used to control access to managed objects in the network. A value of public allows any requestor to access the network resource. The community_name can be up to 32 characters in length and is case-sensitive.

ipaddr
Consists of dotted decimal IP addresses up to 15 characters including dots. They must be in the 256.256.256.256 format. You cannot use any alphabetic characters.

hostname
Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. Hostname is case sensitive. The following is an example of a hostname:

mrpres.whitehouse.capital.usa

Usage Notes:
1. You can include the ipaddr, hostname, or both on the community_name statement.
2. The community name sent by the requester node and the community name expected by the receiving node must match for the request to be executed as requested. If they do not match, the receiving node will use a community name of public by default. Mismatched community names cause authentication errors.

CPCPSESS (SNA)
The CPCPSESS control file entry identifies APPN® control-point to control-point sessions that are to be actively monitored by AON/SNA.

CPCPSESS alias=cp_name,CP1=cp_name,CP2=cp_name

alias
The 1 – 8-character name used to identify the session. Code the alias name in an ACTMON control file entry. Choose names to avoid conflicts with real resources.

CP
The name of the control point. You can qualify the name for a network, for example, network_name.cp_name.

Usage Notes:
- If you are not sure of the control points in your network, use the SNA Automation: APPN Control Points Display panel from the AON/SNA operator interface.
- The session must also have an ACTMON control file entry.

Examples:

This example identifies the control points on three sessions. An ACTMON statement for these entries is also required if these sessions are actively monitored.

CPCPSESS CPCPS1,CP1=TA1CP207,CP2=USIBM.TA01
CPCPSESS CPCPS2,CP1=USIBM.TA1CP208,CP2=USIBM.TA01
CPCPSESS NOWAY,CP1=WAYNE.GARTH,CP2=USIBM.TA01
Use the DDFGENERIC control file entry to define which AON message processing fields are to be used for the DDFADD generic value. DDFADD is used in the DDF panel and DDF tree definitions. By defining a generic DDF entry, you can group DDF notifications as appropriate for your environment.

The DDFGENERIC control file entry is required to implement DDF.

```
DDFGENERIC

func
field,VALUE=(val1,val2,valn)

,LEVELUP=tree_level

,OTHER=value
```

Usage Notes:

When AON issues a notification, each field in the ENVIRON DDF,DDFGENERIC entry is compared with the DDFGENERIC entries. If AON finds a DDFGENERIC definition for an ENVIRON DDF,DDFGENERIC field, it compares the value in the ENVIRON DDF,DDFGENERIC field with the data in the VALUE parameter of the DDFGENERIC entry. If AON finds a match, it uses the DDFGENERIC value. If AON does not find a match, it uses the value in the OTHER parameter of the DDFGENERIC entry. If OTHER is not defined for this field, no value is saved for the field in DDF.

If the ENVIRON DDF,DDFGENERIC field is defined under a specific component and under AON, the component-specific definition is used first, then the general definition is used. A message can be saved for every match found.

Examples:

In the following example, a DDF tree is defined in the EZLTREE member. Refer to this tree when reviewing the following DDFGENERIC examples.

```
/* NETWORK : CNM01 */
1 CNM01
  2 SYSTEM
    3 GATEWAY
    3 GROUPS
      4 CALIF
      5 LA
      5 SANFRAN
      5 SANDIEGO
      4 NEWYORK
      4 ATLANTA
    3 OPID
      4 OPER1
    3 NETWORK
      4 RESOURCE
        4 SNA
        5 SA
        6 NCP
        6 LINE
        6 LINKSTA
        6 CDRM
```

Administration Reference
Examples:

The definitions in the following example of a DDFGENERIC entry cause a PU to be saved on the DDF PU screen:

```
DDFGENERIC AON.RESTYPE,VALUE=(NCP,LINE,LINKSTA,PU,LU,CDRM,CDRSC,APPL),
   OTHER=RESOURCE,LEVELUP=NETWORK

DDFGENERIC AON.RESTYPE,VALUE=(GATEWAY),LEVELUP=SYSTEM
```

If a resource type other than those listed is found, it is saved on the DDF RESOURCE panel, which is defined by the OTHER keyword. LEVELUP=NETWORK causes AON resources to be deleted at the NETWORK level. If a resource type was saved under different generics, all instances are deleted. The LEVELUP definition shows a higher level in the DDF tree than any of the elements under the VALUE parameter.

DDFGROUP

Use the DDFGROUP control file entry to group DDF resources for display purposes to suit your requirements. For example, you can group dissimilar DDF resources by geographic location.

```
|-------DDFGENERIC groupname---,LIST=(res1,res2,resn)--,LIST=(res1,res2,resn)------|
|-------,LIST=(res1,res2,resn)--,LIST=(res1,res2,resn)--------------------------|
```

**groupname**

The name for this group of DDF resources.

**LIST**

The list of resources that you want in the group. Wildcards are supported in resource names. You can create multiple lists; each LIST must be on a separate line.

**Usage Note:**
DDFGROUP

The group name must be in the EZLTREE member and should be on a DDF panel.

Example:

DDFGROUP SANFRAN, LIST=(SASF*, GWATSF, GWFSFD)

ENVIRON AIP

Use the Automation in Progress (AIP) operator status setting in RODM to notify NetView management console users that automation is attempting to recover a failing resource. The status setting is displayed in NetView management console views, thereby preventing a NetView operator from attempting recovery when automation is already working on the problem.

The ENVIRON AIP control file entry contains statements that control which resource types qualify for AIP processing.

LINEPORT
The line or port resource type

PULINKSTA
The physical unit (PU) or link station (LINKSTA) resource type

CDRM
The cross-domain resource manager resource type

NCP
The NCP resource type

Usage Notes:
• If a ENVIRON AIP entry is coded, only those entries specified with Y are enabled.
• If no ENVIRON AIP entry is coded, all resource types are enabled.

Example:

The following are default settings shipped in the ENVIRON AIP control file:

```
ENVIRON AIP, LINEPORT=Y,
   PULINKSTA=Y,
   CDRM=Y,
   NCP=Y,
```

Example:

The following example show how the user could extend the resources available for AIP processing.
ENVIRON AIP, LINEPORT=Y,
    PULINKSTA=Y,
    CDRM=Y,
    NCP=Y,
    AS400=Y

Note: Additional user-written customization would be required to set AIP operator status for AS400.

ENVIRON CONSOLE

The ENVIRON CONSOLE control file entry defines the MVS route codes for AON notifications. The route codes are used when the AON notifications are sent to the MVS consoles as write-to-operator (WTO) messages. This occurs when notification operators are not signed on to NetView or when the special notification operator (SYSOP) is defined. The notification operators are defined in the NTFYOP control file entry.

The ENVIRON CONSOLE entry is optional. The defaults indicated in the syntax are used if no other parameters are specified.

---ENVIRON CONSOLE---

<table>
<thead>
<tr>
<th>ROUTCDE=</th>
<th>codes</th>
<th>ID=</th>
<th>con</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000</td>
<td>01</td>
<td>con</td>
<td></td>
</tr>
</tbody>
</table>

ROUTCDE

Defines the MVS route codes for the message to be displayed. This parameter is a series of on (1) and off (0) characters, representing the routing codes in order. For routing codes, refer to MVS Systems Programming library. The default AON routing code specifies Master Console Action.

ID

Defines the logical console ID to which this message is routed. This routing is performed in conjunction with the routing codes. The default ID is 01. The logical console ID can be determined in the response from the MVS display console command (DC).

MCSFLAG

Defines the MVS MCSFLAG codes. This parameter is a series of on (1) and off (0) characters, representing the flag codes in order. The MCSFLAG codes correspond to MCS flag names that are documented in the MVS Systems Programming Library: System Macros and Facilities. The default AON MCSFLAG code specifies Route conditionally to the console specified in the ID=statement.

Usage Notes:

On a distributed system, AON displays messages on the system console when at least one distributed notification operator is defined and that operator is not logged on. To stop messages from being displayed on the distributed system console, either:

- Do not define distributed system notification operators.
Define an automation operator as a distributed notification operator. The automation operator is always logged on to receive the notification.

If the MCSFLAG parameter is specified as zero, the messages are displayed using the route code. The console ID specification is not used when MCSFLAG=0 is specified.

Example:

In the following example, the route MCSFLAG codes have the default values specified. The ID parameter, which specifies the console to conditionally receive the message, has been changed to a logical console ID of 2. This is used only for EZL603A WTOR during AON initialization.

```
ENVIRON CONSOLE,ROUTCDE=1000000,
   ID=02,
   MCSFLAG=1
```

ENVIRON DDF

The ENVIRON DDF control file entry defines status update characteristics for the environment. The ENVIRON DDF entry is required to implement DDF. Valid values for ENVIRON DDF are listed following the `func` keyword on DDFGENERIC control file entry (see "DDFGENERIC" on page 222).

```
ENVIRON DDF=NO,DDF=(YES),DDFREFRESH=Y
```

DDF

Specifies how the display is to be updated. Following are the values you can use with this parameter:

**YES**

Colors are determined by status. All resources logged under Network generic DDF entry.

**NO**

Do not initialize or log events to DDF.

**STATUS**

DDF colors are defined by resource automation or VTAM status in DDF entries of the control file. Resources are logged under AON resource type generic DDF entries.

**TYPE**

DDF colors are defined by AON resource types in DDF entries of the control file. All resources are logged under Type DDF entry.

**DDFREFRESH**

Defines whether a program is to run each time DDF is started. This is a different program for each AON component. Specifying Y for the
DDFREFRESH parameter primes DDF with the current status of the network resources. Subsequent updates to DDF dynamically take affect as automation events occur.

If DDFREFRESH=Y, all resource statuses are solicited. For a large system, this may be a performance consideration. For AON with AON/SNA, NETSTAT can be run independently with DDF=Y to manually initiate this.

**DDFGeneric**
Specifies which of many possible parts of the message are used to determine how it affects DDF. Valid values include:

- **fromuser**
  The User ID which caused this message to be issued.

- **fromdomain**
  The domain from which this message came.

- **function**
  The AON function that issued this message.

- **status**
  The status associated with the message.

- **resname**
  The resource name.

- **restype**
  The resource type.

- **sp**
  The service point name.

- **opid**
  The operator ID processing this message into DDF.

- **domainid**
  The domain ID the message is from.

- **opt1...opt9**
  The optional parameters 1 to 9. These can be any user-defined values.

The message processor determines whether these field names contain valid DDF generic values. If multiple fields are defined, the potentially DDF descriptors can be saved under multiple generic values. (See [DDFGENERIC on page 222](#).)

At least one DDFGENERIC statement should be defined for each field specified in this parameter.

**DDFAuto**
Specifies whether DDF updates occur for all resources (N), or for only those eligible for automation (Y). The default is N.

**DDFMark**
Specifies whether the MARK on a resource will be saved, when the resource in the DDF is updated, if the operator is not logged on. Following are the values you can use with this parameter:

- **RESET**
  The MARK is not saved if the operator that marked the resource is logged off when the resource is updated.

- **KEEP**
  The MARK will be saved regardless of the status of the operator that marked the resource.

**Example:**
ENVIRON DDF

The following example specifies that the colors are determined by status, the DDF program will not run, and the resource type and service point parts of the message to be used.

```
ENVIRON DDF, DDF=Y,
    DDFREFRESH=N,
    DDFAUTO=N,
    DDFGENERIC=(RESTYPE,SP),
    DDFMARK=RESET
```

In this example, resource type and service point name fields in AON notifications are checked against DDFGENERIC statements. If a match is found, the value of the field is used as the DDFGENERIC. Also, DDFMARK=RESET resource is updated after an operator marks the resource and logs off; the mark on the resource will be lost.

ENVIRON EXIT

Commands you define in ENVIRON EXIT control file entry are issued by AON when it calls the associated common routines. Using this exit interface, you can call your user-written command lists when one of the common routines runs. You can use exits to change global variables set by routines or to run site-specific procedures.

```
ENVIRON EXIT

EXIT05 = ("command")
EXIT06
EXIT07
EXIT08
EXIT09
```

EXIT05

This parameter defines default exit call values for the EZLEAGRN common routine. When common routine EZLEAGRN is called, AON calls the routines defined here when VTAM resource information is gathered.

EZLEAGRN is called when AON gathers resource information from VTAM.

- **command**
  - Specifies the exit to be run or the command to be issued.

- **parms**
  - Specifies the parameters to be passed to the exit or the command to be issued.

EXIT06

This parameter defines default values for the EZLEATHR common routine. The EZLEATHR common routine is called whenever AON checks thresholds.

Unless exit values are specified on the THRESHOLDS control file entry for a resource.

The EZLEATHR common routine is called when AON checks thresholds.
EXIT07
This parameter defines default values for the EZLECAUT common routine. Unless values are specified on the RECOVERY control file entry for a resource, AON issues the command defined here when the EZLECAUT common routine is called to check recovery flag values.

The EZLECAUT common routine is called when recovery definitions are evaluated to set automation flags.

EXIT08
This parameter is run when messages are issued. Using this parameter, a user-written routine can translate, add new message classes, or use and update the message text.

When called, the user-written routine has resource name or resource type in the order coded in the EXIT08 statement.

Information coded on the EXIT08 statement is preserved and passed to the exit.

EXIT09
EXIT09 can be used to change the way AON recovers a SNA resource, or class of resources by setting the TGLOBALs documented. EXIT09 can be used to perform some function which is desirable each time a recovery interval is processed, but not necessarily change the AON recovery logic.

Attention: The TGLOBALS contain data that guides AON recovery processing. Indiscriminate changes to the TGLOBALS can cause unpredictable results.

Usage:
You can code multiple exit lines, such as EXIT05, as shown in the examples. When multiple EXIT0n lines are coded, AON performs the specified actions sequentially.

For the exits to pass global variable names, an ampersand (&) must precede the valid global variable name. For example, to pass RESNAME, code &RESNAME in the exit.

When you run multiple exits, exit processing stops when it encounters a nonzero return code. You can change the return code from either EZLEAGRN, EZLEATHR, or EZLECAUT by setting the following TGLOBALS from user exits:
EZLEAGRN EZLEZLEAGRNRC
EZLEATHR EZLEZLEATHRRC
EZLECAUT EZLEZLECAUTRC

You must set the return code to a numeric value between 0 and 9. If the TGLOBAL contains anything else, the routine returns a return code of 99.

ENVIRON RACF
The ENVIRON RACF control file entry specifies that the same RACF data set is shared among different NetView domains. This entry is used by AON routines to coordinate and automate RACF password maintenance for gateway automation operators. This control file entry is optional.
ENVIRON RACF

ENVIRON RACF

\[ \text{OWNER=owner\_domain} \]

\[ \text{SHARE=(share\_domain,share\_domain,share\_domain)} \]

\[ \text{LIST=(owner\_domain,share\_domain,share\_domain)} \]

\[ \text{MASK=%\text{racf\_value}} \]

OWNER
In systems with a single shared RACF environment, OWNER identifies the NetView domain name that maintains RACF passwords for gateway automation operators.

This domain is used when building the VSAM key to retrieve and update the RACF passwords. The passwords are stored in encrypted format on a VSAM data set using a key of domain name and operator ID.

SHARE
In systems with a single shared RACF environment, SHARE identifies NetView domains that share the same RACF data set with the owning domain, specified on the OWNER parameter.

LIST
In systems with multiple shared RACF environments, LIST identifies the owning domain and any shared domains. You can code multiple LIST parameters, as shown in the examples.

MASK
If your installation has specific guidelines for passwords, you can implement those with the MASK option. AON supports a maximum of eight characters for passwords. MASK defines which characters are valid in the generated password. If specified, AON generates new RACF gateway passwords every 30 days.

In MASK definitions, two characters in the entry define a single character in the password. MASK definitions can be a maximum of 16 characters to define an eight-character password.

The first character in each pair defines whether the second character of the pair is a type or is fixed. The first character is defined by using one of the following symbols:

\%
Next character is the type of password character.

\$
Next character is a fixed password character.

The second character in the pair identifies the type or the fixed character as follows:

\(A\) Alphabetic
\(V\) Vowel
\(C\) Consonant
\(W\) No vowel
\(X\) Alphanumeric
\(N\) Numeric

Usage Notes:
The OWNER and SHARE parameters are mutually exclusive with the LIST parameter.

MASK is a system-wide parameter. MASK values must be valid for any system to which this domain can log on.

Example:

In this example, NetView domains CNM01, CNM02, CNM04 and CNM99 share the same RACF data set. CNM01 is the domain responsible for automated maintenance of RACF passwords for gateway automation operators in all the above domains.

```env
ENVIRON RACF,OWNER=CNM01,
  SHARE=(CNM02,CNM04,CNM99),
  MASK=@(#) 89 1.31@(#)N$Z%N
```

The MASK keyword defines the 4-character password convention. The first character is alphanumeric, the second is numeric, the third is Z, and the fourth is numeric.

Example:

The following example defines an environment with three sets of domains with shared RACF. The owning domains (CNM01, CNM11, and CNM21) maintain gateway passwords for the domains that follow them in their respective LIST definitions.

```env
ENVIRON RACF,LIST=(CNM01,CNM02,CNM03,CNM99),
  LIST=(CNM11,CNM12,CNM13,CNM19),
  LIST=(CNM21,CNM22,CNM23,CNM29)
```

---

**ENVIRO SETUP**

The ENVIRO SETUP control file entry identifies attributes of the NetView program that is running AON and its supporting operating system. Use the ENVIRO SETUP entry to tailor the control file.

The ENVIRO SETUP entry is optional. If you do not include it, AON uses all of the defaults.

```
environ setup
  ,NETVIEW=NET
  ,HELDMSG=(type,type,type)
  ,SYSNAME=name
  ,GENALERT=Y or YES
  ,FTAFPFX=xxxxxx
```
NETVIEW

NET Specifies that this is an AON environment.

HELDMSG

Specifies the type or types of messages to be held on the operator’s workstation. The message types are:

I or INFO

Informational messages

W or WARN

Warning messages

E or ERROR

Error messages

A or ACTION

Action messages

You can override these values in the AUTOOPS statement for each operator.

SYSNAME

Defines the system from which messages are forwarded and as the root component name for DDF. The value for this parameter must match the level 1 tree name in the EZLTREE DSIPARM DDF customization member.

GENALERT

Defines whether resource-related AON notifications are to be logged in the hardware monitor database as alerts. Y is required to generate problem alerts and resolutions. The default is N which improves overall NetView performance.

To update the NetView management console (NMC) and the Resource Object Data Manager (RODM) with automation information, code GENALERT=Y.

FTAFPFX

Defines the first six characters of the full-screen TAF Access Method Control Blocks (ACBs) defined in the NetView AON ACB definitions. FTAFPFX is used by the AON TAF command facility to start TAF sessions through a full-screen command facility (OP). This is not used by the NetView BFSESS command, which requires a naming convention for these ACBs of TAFnnn, where nnn is the last three characters of the domain ID.

The FTAFPFX parameter is required only if you are using the FULLSESS entry. The default is TAFxxx (last 3 characters of the domain ID).

• When FTAFPFX is not defined, code the following TAF ACB:
TAFxxFyy

Where:
xx Last 2 characters of the NetView domain ID, such as CNM01
yy Last 2 characters of the operator logon APPLID, such as CNM01000

TAF SRCLU = TAF01F00

- When FTAFPFX is defined, code the following TAF ACB:
  xxxxxxxyy

  Where:
  xxxxxx Value specified in FTAFPFX parameter TAF SRCLU = TAFANF00
  yy Last 2 characters of the operator logon APPLID, such as CNM01000

Supports a NetView domain ID of less than 5 characters.

Note: You should define an equal amount of TAF ACBs as the number of your NetView operators.

DBMAINT
Specifies how the status file is allocated. The value is either REUSE or EXPORT. The default is EXPORT. The value specified for DBMAINT must match the way the VSAM data sets were allocated or errors occur.

RODMAIP
 Defines whether the RODM AIP (Automation In Progress) operator status is set, causing the AIP pattern to display for affected resources in NetView management console.

RODMNAME
The name of the RODM where the AIP operator status and Operator Intervention View (OIV) are set. This field is required if RODMAIP is set to YES.

Refer to the usage notes in the AONAI P command in the Tivoli NetView for z/OS Command Reference for information regarding the RODM user ID.

RODMDOM
If the domain where RODM and NMC information is gathered is different from the current AON domain, include the target domain. The default for the RODM domain is the current AON domain.

INFORMPM
Defines the DSIPARM member that contains the INFORM policy to be loaded during initialization. Use the INFORMTB command to load the inform policy member manually.

Usage Notes:

If HELDMSG is not specified on NTFYOP entries or the ENVIRON SETUP entry, the default is to roll all messages. NTFYOP HELDMSG parameters override the defaults specified in the ENVIRON SETUP control file entry. For example, specifying HELDMSG=(W,I) on the ENVIRON SETUP entry and HELDMSG=E on the
**ENVIRON SETUP**

NTFYOP entry result in only error messages being held for that operator. However, if HELDMSG is not specified on the NTFYOP entry, the ENVIRON SETUP default is used.

Specify only RODMDOM in the ENVIRON SETUP list when the domain of your RODM is different than the current AON domain. AON uses the current domain for RODMDOM as the default. If you specify a RODM domain that is not valid AIP operator status updates will be misdirected.

The Operator Intervention View (OIV) is created when you specify RODMOIV=YES and RODMAIP=YES in the ENVIRON SETUP control file entry. AON will attempt to create the view during initialization, or later, when AON attempts to populate the view.

AIP and OIV functions will use the RODM user ID domainid concatenated with AON when accessing RODM.

**Example:**

Following is an example of an ENVIRON SETUP control file entry:

```
ENVIRON SETUP, NETVIEW=NET,
    GENALERT=YES,
    FTAFPFX=TAFA1F,
    SYSNAME=NTV6D,
    DBMAINT=EXPORT,
    ROdmaip=YES,
    RODmoiv=YES,
    RODMNAME=RODM1,
    HELDMSG=(INFO, WARN, ERROR, ACTION)
```

**ENVIRON SNBU (SNA)**

The ENVIRON SNBU control file entry specifies to AON/SNA the conditions under which switched-network backup automation is to occur for the LPDA-2 capable modems that have definitions in the control file.

**ENVIRON SNBU**

CRIT

Initiates switched network backup automation when critical thresholds resource errors has been reached (PU or LINE). Use FKVSNBUCRIT to set the common global variable.

MONIT1

Initiates AON/SNA SNBU automation when the first MONIT interval has been run. Although one reactivation recovery cycle has occurred, AON/SNA did not successfully reactivate the resource. Use FKVSNBUMON1 to set the common global variable.
LEVEL
Initiates AON/SNA SNBU automation when either Level 1 or Level 2 connected modems are supported by this environment. Use FKVSNBULVL to set the common global variable.

DELAY
Specifies how many seconds to wait for NCP/VTAM before continuing to the next steps in AON/SNA SNBU automation. Use FKVSNBUWAIT to set the common global variable.

Usage Note:
The ENVIRON SNBU control file entry is required to perform SNBU automation.

Example:
In this example, AON/SNA authorizes AON/SNA SNBU automation to run when critical threshold failure levels are reached:
ENVIRON SNBU,Crit=Y,MONIT1=N,LEVEL=1,DELAY=5

ENVIRON TIMEOUT

The ENVIRON TIMEOUT control file entry defines the maximum allowable time limits for certain operations in the control file. If the time limits are exceeded, the control file considers them error conditions and takes appropriate action.

The ENVIRON TIMEOUT entry is optional. If you do not include it, AON uses the defaults.

Usage Notes:
AON adds a suffix of TIME to all keyword names and creates CGLOBAL names for them. The names created by AON and its components are:
• DIALTIME
• RCMDTIME
• PINGTIME
• ACTIVETIME
• SNMPTIME
• RETRYTIME
ENviron Timeout

**Note:** To set WAITTIME and XDOMTIME, see the CNMSTYLE sample. You cannot perform that task by using ENVIRON TIMEOUT.

These names can be used in user-written programs if they are properly specified.

Users can specify additional timeout names to add environment-unique specifications. The routine that generates the CGLOBALs processes any name, as long as the name of the time parameter is six characters or less in length. Edit checking for the values between 1 and 32000 occurs.

If a timeout value greater than 29 seconds is specified, NetView may echo unsuppressible warnings about the WAIT statement to the NetView console. This does not impact AON operation.

**Example:**

An AON entry example:

```
ENVIRON TIMEOUT,DIAL=120,
    RCMD=90,
    PING=90,
    ACTIVE=5
    RETRY=3
```

In this example, `ACTIVE=5` directs EZLEVACT to wait five seconds for VTAM to respond to a request to activate. All other specifications are defaults.

**Example:**

The following is an AON entry with a user-defined wait time between PING retry attempts:

```
ENVIRON TIMEOUT,DIAL=120,
    RCMD=90
    ACTIVE=29
    PING=90
    RETRY=3
```

In this example, `RETRY=3` is the number of seconds to wait between PING RETRY attempts.

**Attention:** Setting this number too large could cause the active monitoring process to slow down due to excessive wait time. Active monitoring will perform up to 3 pings and exit upon completion of the first successful ping. If the node is unknown after the third ping, active monitoring initiates EZLEFAIL processing.

---

**EZLTLOG**

The EZLTLOG control file entries specify the commands that process AON automation logs when the logs become full. These entries allow the log switching and the jobs that copy the AON log to a history file to be tailored to the naming conventions of your installation. EZLTLOG is a required entry.

```
EZLTLOG PRIMARY
    AUTOFLIP=YES
    LIT='literal'
    JOB='cmd'
```
Specifies whether this is the primary or secondary log. If NONE is specified, EZLTLG is not initialized and no NLOG logging is performed. Specifying NONE may improve performance on some systems by reducing input and output. All records are filed in the NETLOG.

**AUTOFLIP**

Specifies whether the log is to switch automatically when it becomes full. The default is NO.

**LIT**

Specifies a literal string that is used in messages EZL307, EZL308, and EZL309. The messages are sent out as AON notifications when the log switch process is performed.

**JOB**

Specifies the job in proclib to submit a job to archive the primary or secondary automation log.

**Usage Notes:**

The subsystem interface (SSI) must be running. Jobs EZLSJ007 and EZLSJ009 must also be in an active procedure library.

You can change jobs or member names that define logs. Primary and secondary entries should be supplied.

The AON sample library contains the procedure and the IDCAMS control statements necessary to perform switch and off-load. Jobs EZLSJ007 and EZLSJ009 copy the VSAM data sets into a sequential history log and clears the AON logs. When the logs are clear, NetView can open the files as output data sets.

**Examples:**

In the following text USER.PROCLIB is the partitioned data set where the job resides.

In this example, the primary and secondary logs are defined. The logs are automatically switched when they are full. The command SUBMIT USER.PROCLIB(EZLSJ007) is issued to process the primary log and SUBMIT USER.PROCLIB(EZLSJ009) is issued to process the secondary log:

EZLTLG PRIMARY,JOB=USER.PROCLIB(EZLSJ007),AUTOFLIP=YES,
  LIT='AUTOMATION LOG'
EZLTLG SECONDARY,JOB=USER.PROCLIB(EZLSJ009),AUTOFLIP=YES,
  LIT='AUTOMATION LOG'

**Examples:**

In the following example, the primary and secondary logs are switched automatically when they become full. No job is designed to archive historical data in the history file.

EZLTLG PRIMARY,AUTOFLIP=YES
EZLTLG SECONDARY,AUTOFLIP=YES
The FORWARD FOCALPT control file entry identifies NetView domains to which the automation notification messages are to be forwarded. This entry defines both a focal point and a backup focal point.

When automation notification forwarding is used, define this control file entry at the distributed hosts. This entry is required only if you implement focal points. If the FORWARD FOCALPT entry is not included, AON uses the defaults.

PRI
Identifies the NetView domain name to which automation notifications are forwarded. Define this domain as a focal point. If you do not specify a domain, AON the NetView domain name where AON resides is the default. If AON defaults, it considers this NetView as a focal point and displays all automation notification messages without forwarding them.

When automation notification forwarding is not required, use the default or specify your domain name as the focal point name.

BKUP
Identifies the NetView domain name to which the automation notifications are forwarded when the focal point is not available. Define this domain as a focal point.

If you do not specify a domain, AON uses the NetView domain name (default) under which the control file initializes.

If you specify the entry at the primary focal point, use the default or specify the primary focal point domain name.

When automation notification forwarding is not required, use the default or specify your domain name as the focal point name.

Example:

This example defines a focal point domain for a single distributed host, where CNM01 is the primary focal point and CNM03 is the distributed host. In this example, domain CNM01 is the focal point to which automation notifications are forwarded. The distributed entry contains the line:

```
FORWARD FOCALPT,PRI=CNM01
```

The entry for both the focal point and distributed host also need a gateway definition. See "GATEWAY" on page 239.
The FULLSESS control file entry identifies applications with which AON operators can establish full-screen AON TAF sessions automatically using the operator interface.

\[ \text{FULLSESS name}, \text{APPLID=appl}, \text{SYSTEM=sys} \]

**name**
Describes the application for which a full-screen TAF session is to be established. This field is displayed as the description field on the AONTAF operator interface menu. It can be the same as the entry coded in the APPLID field.

**appl**
The application name specified to VTAM.

**sys**
The system name on which the application runs. System is an information only entry that is displayed on the AONTAF operator interface menu.

**Usage Notes:**
Include a separate FULLSESS entry in the control file for each application. This enables usage for all operators.

If FTAFPFX is specified in the ENVIRON SETUP entry, the ACB is \( xxxxxxnn \), where \( xxxxxx \) is the TAF prefix defined and \( nn \) is the last two characters of the NetView operator ACB.

Use the following format for the VTAM ACB:

**NetView logon ACB name**
\( xxyynnn \), where \( xxyy \) is the 5-character domain ID and \( nnn \) is the ACB number (000–099)

**TAF logon ACB name**
\( \text{TAFyyFnn} \), where \( yy \) is the last 2 characters of the domain ID and \( nn \) is the ACB number (00–99)

The number TAF ACBs should equal the number of NetView operators.

**Example:**
Following is an example of a full-screen entry for TSO.
\[ \text{FULLSESS TSO,APPLID=TA1N1} \]

**GATEWAY**
The GATEWAY control file entry identifies NetView domains to which AON routes automation notification messages. When automation notification forwarding is used, this control file entry is defined at the distributed and focal point hosts.
When you define a focal point hierarchy in the distributed nodes for automation notification, at least one GATEWAY entry must point to the next higher domain in the hierarchy to enable proper routing to the focal point (see “FORWARD FOCALPT” on page 238).

Gateway entries must also be defined in the control file of the focal point so that the focal point can re-establish the connection to the distributed nodes when the focal point host is recycled. Additional gateway entries can also be included for allowing backup to focal points through intermediate hosts.

**Note:** Only NNT sessions are support by message routing.

```
GATEWAY domain, PASSWORD=RACFNNT pw, LOGONID=GATOPER gate
```

domain
Defines the NetView domain to which an NNT session is established (to which automation notifications are routed).

**PASSWORD**
Defines the password used when the local gateway operator establishes an NNT session with the domain defined in this entry.

RACFNNT specifies that the RACF password for the outbound gateway automation operator to log on to the specified domain-name is to be managed and retrieved by AON.

If RACFNNT is not specified, specify a password for the outbound gateway automation operator.

**LOGONID**
Defines the user ID for the inbound gateway operator at the domain described in this GATEWAY entry. Ensure that this user ID is defined in DSIOPF on the target domain and that no other domains are using this gateway ID. If GATOPER is specified, AON generates the gateway name GATdomain.

**Note:** To run both AON and SA/390 in the same NetView domain, you must customize your GATEWAY policy definitions. Only one product can use GAT||domai inas the task name; therefore, to customize the AON Gateway tasks, choose a unique task name for each GATEWAY LOGINID=gate, where gate is the task name.

**LOGMODE**
Specifies VTAM session parameters. These are used when the START domain command is issued. If LOGMODE is not defined, no logmode is sent and the VTAM default is used.

**DESC**
Describes the NetView domain.

**Usage Notes:**
The PASSWORD=RACFNNT parameter cannot be used without tailoring NetView to activate it.

Include a gateway entry for each domain with which this NetView communicates directly. For the notification forwarding function to operate correctly, a two-way communication pipe must be established. For more information about the setup and use of message forwarding, see "FORWARD FOCALPT" on page 238.

**Example:**

The following example defines a gateway entry without RACF password protection:

```plaintext
GATEWAY CNM02,PASSWORD=AUTOOP1,DESC='CNM02 NetView'
```

The current domain is CNM01. In this example, the password for the gateway automation operator (GATCNM01) to log on to domain CNM02 from domain CNM01 is AUTOOP1. GATCNM01 is the default name for the outbound gateway automation operator for domain CNM01 and inbound gateway automation operator for domain CNM02.

**Example:**

The following example defines a gateway entry with RACF password protection:

```plaintext
GATEWAY CNM02,PASSWORD=RACFNNT,DESC='CNM02 NetView'
```

This example is different from Example 1 in that AON retrieves and supplies the RACF password for GATCNM01 to log on to domain CNM02.

**Example:**

The following example defines a focal point domain for a single distributed host, where CNM01 is the primary focal point and CNM03 is the distributed host. The primary focal point entry points to the distributed host.

```plaintext
GATEWAY CNM03,DESC='DIST. HOST',PASSWORD=PWD3
```

The distributed entry is:

```plaintext
GATEWAY CNM01,DESC='DIST. HOST',PASSWORD=PWD1
```

NetView domain CNM01 is the normal focal point to which automation notifications are forwarded.

**Example:**

The following example defines an intermediate host (CNM02), and distributed host (CNM03). The primary focal point entry points to the intermediate host.

```plaintext
GATEWAY CNM02,PASSWORD=PWD2
ADJNETV CNM02,DOMAIN=CNM02
```

The intermediate host entry is:

```plaintext
GATEWAY CNM01,PASSWORD=PWD1
GATEWAY CNM03,PASSWORD=PWD3
```

The distributed host entry is:

```plaintext
GATEWAY CNM02,PASSWORD=PW3
ADJNETV CNM01,DOMAIN=CNM02
```
All automation notifications from the distributed host are forwarded to CNM01 through CNM02, which is defined as an adjacent NetView. A backup focal point has not been defined for the distributed host.

Note: ADJNETV entries are explained in "ADJNETV" on page 214.

**INSTALLOPT**

The INSTALLOPT control file entry is used to define AON or one of its automation components. Code an INSTALLOPT statement for AON, each of its components that you install, and any automation you want to add to AON. Not all keywords are valid for all automation components.

```
INSTALLOPT component, INITIALIZE= [Y N]
```

```
, DEFTAB= option_definition_table, IC= initial_clist
```

```
, OPER= autoops_id, FAILOP= autoops_id, RECOVOP= autoops_id
```

```
, MSGOP= autoops_id, ACTMONOP= autoops_id
```

**component**

Defines the component installed, for example, specify AON for AON:

- AON
- APPN
- IP390
- NVAIX
- SA
- SNA
- SNBU
- TCPIP
- X25

**INITIALIZE**

Defines whether to initialize the component when AON starts. The default is Y.

**DEFTAB**

Defines the option definition table that is used for initialization parameters. EZLTABLE is the option definition table for AON.

**IC**

Defines the program that initializes the options environment. These definitions include tasks such as loading tables, starting automation operators, and starting active monitoring timers.

**OPER**

Defines the automated operation that runs the initial program. Define an AUTOOPS statement for this operator ID.
**FAILOP**
Directs failure processing to run under a particular automated operation. This processing includes running the EZLEFAIL program and subsequent programs coping with resource failures and recovery monitoring. Define an AUTOOPS statement for this operator ID.

**RECOVOP**
Directs recovery processing to run under a particular automated operation. This includes running the EZLERECV program and subsequent programs coping with resource recoveries. There should be an AUTOOPS statement for this operator ID.

**MSGOP**
Directs messaging to run under a particular automated operation. This includes sending messages to operators, logs, and DDF. Define an AUTOOPS statement for this operator ID.

**ACTMONOP**
Directs active monitoring to run under a particular automated operation. Define an AUTOOPS statement for this operator ID.

**Usage Note:**
To install and initialize automation, define an INSTALLOPT statement.

**Example:**
The following default entry uses the option definition table EZLTABLE to start AON:

```
INSTALLOPT AON,INITIALIZE=Y,DEFTAB=EZLTABLE
```

**Example:**
In the following example, a user-written automation component, referred to as USER, is started. USRTABLE is the option definition table used to start USER.

```
INSTALLOPT USER,INITIALIZE=Y,DEFTAB=USRTABLE
```

**Example:**
The following example defines the MVS TCP/IP function. INSTALLOPT uses the existing INSTALLOPT parameters.

```
INSTALLOPT IP390,INITIALIZE=Y,FAILOP=TCP3OP2,RECOVOP=TCP3OP2,MSGOP=TCP30PM
```

**Note:** For parameters that are common with the INSTALLOPT NV6000 Statement, use the INSTALLOPT NV6000 definition for input.

---

**IPCONN**
The IPCONN control file entry defines thresholds to be used when monitoring connections for a given application.
proc_name
The name of the procedure associated with the application. Wildcards (*) are supported.

stack_name
The name of the stack to use to monitor connections with this application. This should match a TCP390 policy definition name.

THRESH
The thresholding information:

byte_count
The number of bytes output at the MVS host.

type
The type of threshold for byte_count. Valid values are IDLE, MIN, MAX, or an asterisk (*). The asterisk is equivalent to specifying IDLE.

interval
The threshold for the amount of time a session can be inactive, in the form hh:mm:ss.

action
Valid action types are as follows:

NONE
No action is taken. NONE cannot be specified with another action. NONE is the default

NOTIFY
Sends a notification to a specified user or task using the definitions set in the AON notification policy.

DROP
Breaks the connection and logs the appropriate message if NOTIFY is not specified.

ACTMON=IPCONN
This is required to reference the timer interval in order to prevent more than one timer for all IPCONN definitions.

Usage Notes:
• When an asterisk (*) is specified for byte_count, the type is ignored.
• Multiple THRESH statements are supported. The first one in the list that is matched will use the value specified for action.

Example:

The following example specifies two THRESH options:

IPCONN XYZ*,SP=NMPIPL10,THRESH=(*,*,00:04:00,DROP),THRESH=(2147,MIN,00:00:30,DROP),
ACTMON=IPCONN

For all connections with an application or applications of XYZ* associated with stack NMPIPL10, the following will occur:
• If idle for more than four minutes, the connection will be broken.
• If idle for more than 30 seconds, less than 2147 bytes have been sent, and the send window size is zero (0), the connection or connections will be broken and message FKX108I will be issued.
The IPHOST control file entry defines an IP Host to IP390 and optionally defines active monitoring of the host.

AON_Name

The unique name associated with the TCP/IP host that is used by AON/TCP. The name must be a valid REXX symbol.

MVS_Stack_Name

The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

OPTION=IP390

Only valid entry is IP390.

IPADDR

Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 256.256.256.256 format. Alphabetical characters are not valid.

HOSTNAME

Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The host_name variable is case-sensitive. The following is an example of a host_name:

mrpres.whitehouse.capital.usa

INTERVAL

 Defines the monitoring interval in hh:mm format. This is required for proactive monitoring.
ACTMON_Def

Defines active monitoring for groups of resources in the network. For additional information, see "ACTMON" on page 209.

FORMAT

Specifies one of the following options to be used to determine the resource’s status.

PING  Pings the resource to check its status.

SNMP  Uses SNMP MIB polling to check the status of the resource.

STATUS

Defines the expected (AON) status of the resource.

MIBVAR1 – n

Multiple MIBVAR statements can be defined and is needed only for SNMP thresholding (requires FORMAT=SNMP).

CORRELATE Y|N

Set to Y if trap correlation of IPHOST’s/IPROUTER’s/IPTN3270’s and their respective interface is desired. The CORRELATE parameter should only be set to Y for resources when SNMP is available. All known interfaces will be used to correlate the status. Correlation is only recommended for interfaces installed on critical IPHOST’s or IPROUTER’s. N is the default.

EXIT14

Defines additional processing of SNMP interface table. The default value is NONE. Invoked for FORMAT=SNMP.

EXIT15

Defines additional processing of user defined thresholds (MIBVAR). The default value is NONE.

Example:

The following example defines host pkoch for proactive monitoring using SNMP polling. MIB thresholding for pkoch will be done based on MIBVAR1, MIBVAR2, MIBVAR3, and MIBVAR4.

```
ACTMON IP390,OPTION=IP390,INTVL=01:00,STATUS=NORMAL
ACTMON IPHOST,OPTION=IP390,INTVL=00:30

IPHOST PKOCH,SP=NMPIPL10,
    OPTION=IP390,
    HOSTNAME=pkoch.raleigh.tivoli.com,
    INTVL=45
    FORMAT=SNMP,
    STATUS=(NORMAL,TRESH*,DEGR*),
    MIBVAR1=(tcpActiveOpens.0,LT,5000),
    MIBVAR2=(tcpInErs.0,GT,5),
    MIBVAR3=(lplnDiscards.0,EQ,1),
    MIBVAR4=(ipRoutingDiscards.0,GE,3)
```

IPINFC

The IPINFC control file entry defines an IP interface to IP390 and optionally defines active monitoring of the interface.
AON_Name
The unique name associated with the TCP/IP interface that is used by AON/TCP. The name must be a valid REXX symbol.

Stack_Name
The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

OPTION=IP390
Only valid entry is IP390.

IPADDR
Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 256.256.256.256 format. Alphabetical characters are not valid.

HOSTNAME
Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The Host_name variable is case-sensitive. The following is an example of a host name:

mrpres.whitehouse.capital.usa

INTERVAL
Defines the monitoring interval in hh:mm format. This is required for proactive monitoring.

ACTMON_Def
Defines active monitoring for groups of resources in the network. For additional information see “ACTMON” on page 203.

FORMAT
Specifies one of the following options to be used to determine the resource status.

PING  Pings the resource to check its status.

SNMP  Uses SNMP MIB polling to check the status of the resource.
### IPINFC

**STATUS**
Defines the expected (AON) status of the resource.

**MIBVAR1 – n**
Multiple MIBVAR statements can be defined and is needed only for SNMP thresholding (requires FORMAT=SNMP).

**EXIT14**
Defines additional processing of SNMP interface table. The default value is NONE. Invoked for FORMAT=SNMP.

**EXIT15**
Defines additional processing of user defined thresholds (MIBVAR). The default value is NONE.

**Example:**

The following example defines INFC MLRITE as part of the HOST policy grouping to be monitored every 15 minutes by SNMP, checking 1 MIB variable each time. A NORMAL or DEGRADED status is satisfactory:

```
ACTMON IP390,OPTION=IP390,INTVL=01:00,STATUS=NORMAL
ACTMON IPINFC,OPTION=IP390,INTVL=00:30
ACTMON HOST,OPTION=IP390,INTVL=00:15,
    FORMAT=SNMP,
    MIBVAR1=(ipRoutingDiscards.0,GE,3),
    STATUS=(NORMAL,DEGRADED)

IPINFC MLRITE,OPTION=IP390,
    ACTMON=HOST,
    SP=NMPIPL10,
    IPADDR=9.37.74.18,
    HOSTNAME=mlrite.raleigh.ibm.com
```

### IPNAMESERV

The IPNAMESERV control file entry defines an IP NameServer to IP390 and optionally defines active monitoring of the NameServer.

```
IPNAMESERV ---AON_Name---,---SP---stack_name---,---OPTION---IP390---
```

```
IPNAMESERV ---AON_Name---,---SP---stack_name---,---OPTION---IP390---
    ---IPADDR---IPaddr---,---HOSTNAME---host_name---
    [---INTVL---interval---]
```

```
IPNAMESERV ---AON_Name---,---SP---stack_name---,---OPTION---IP390---
    [---ACTMON---Actmon_Def---]
    [---FORMAT---PING---SNMP---]
    [---STATUS---(---status1---,...,statusn---)]
```

248 Administration Reference
AON_Name
The unique name associated with the TCP/IP Name Server used by AON/TCP. The name must be a valid REXX symbol.

stack_Name
The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

OPTION=IP390
Only valid entry is IP390.

IPADDR
Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 256.256.256.256 format. Alphabetical characters are not valid.

HOSTNAME
Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The host_name variable is case-sensitive. The following is an example of a hostname:

mrpres.whitehouse.capital.usa

INTERVAL
Defines the monitoring interval in hh:mm format. This is required for proactive monitoring. If the interval is defined for ISTAT, but not specified, monitoring will not occur.

ACTMON_Def
Defines active monitoring for resources in the network. For more information, see “ACTMON” on page 209.

FORMAT
Specifies one of the following options to be used.

PING
Pings the resource to check its status.

SNMP
Uses SNMP MIB polling to check the status of the resource.

STATUS
Defines the expected (AON) status of the resource.

MIBVAR1 – n
Multiple MIBVAR statements can be defined and is needed only for SNMP thresholding (requires FORMAT=SNMP).

EXIT14
Defines additional processing of SNMP interface table. The default value is NONE. Invoked for FORMAT=SNMP.

EXIT15
Defines additional processing of user defined thresholds (MIBVAR). The default value is NONE.
Example:

The following example defines DNS RAL11 as part of the DNSMON policy grouping to be monitored every 15 minutes through PING. A NORMAL status comes from the IP390 level.

```
ACTMON IP390,OPTION=IP390,INTVL=01:00,STATUS=NORMAL
ACTMON IPNAMESERV,OPTION=IP390,INTVL=00:30
ACTMON DNSMON,OPTION=IP390,INTVL=00:15,
    FORMAT=PING

IPNAMESERV RAL11,
    SP=NMIPL10,
    OPTION=IP390,
    IPADDR=9.37.240.3,
    HOSTNAME=ralname11.raleigh.ibm.com,
    ACTMON=DNSMON
```

IPPORT

The IPPORT control file entry defines IP ports to IP390 and optionally defines monitoring of the specified port.

```
CONTRIBUTION—Socket_Name
    ,SP=stack_name

    ,PORT=Port_Num– | *
    ,TCPNAME=User_ID

    ,PROTOCOL=Port_Type
    ,SESTAT=YES–NO

    ,CMDTYPE=MVS–TSO–UNIX–NETV

    ,DELAY=Mins

    ,ACTIVATE="Start_Command"
    ,STOP="Stop_Command"

    ,EXIT13="User_Exit"
    ,DESC="User_Text"

    ,ACTMON=Actmon_Def
    ,FORMAT=PORT

    ,STATUS=(status1,...,statusn)
```
Socket_Name
The unique name associated with the TCP/IP socket that is used by AON/TCP. The name must be a valid REXX symbol.

SP=stack_name
The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point definition. For additional information see "TCP390" on page 279.

PORT=Port_Num | *
The expected TCP/IP port number. The port numbers are divided into the following groups:
- 0 – 1023 are well known ports
- 1024 – 49151 are registered ports
- 49152 – 65535 are private ports

If an asterisk (*) is coded, then AON dynamically determines what TN3270 sockets are known to TCPIP. This enables you to run multiple TN3270 sockets and requires CS/390 V2R6 or later. Each socket will have a policy name of TN-concatenated with the port number. For example, TN-1055 is used for TN3270 sessions connecting into port 1055. An asterisk is valid only for TN3270 definitions.

Note: When running CS/390 V2R6 or later with multiple TN3270 ports, any changes to the dynamic policy definitions created by AON will not be reset until the next stack monitor interval.

TCPNAME=UserID
The MVS job name of the application that opened the port.

PROTOCOL=Port_Type
Defines the type of socket. The following are two types of ports:
- TCP
- UDP

SESSTAT=
Enables the socket to be defined, but not shown, in the session status panels.

YES When SESSTAT = YES or SESSTAT is not coded, then sessions for that socket will be displayed when invoking IPSTAT.

NO When SESSTAT=NO sessions, if any exist, for that socket will not be available to IPSTAT.

CMDTYPE
Defines the type of command. The following are four types of commands:
- MVS (default)
- TSO
- UNIX
- NETV

DELAY=Mins
Defines the number of minutes the process is delayed after the STOP command is issued. If this is not specified, the ACTIVATE command is issued immediately after the STOP command.

ACTIVATE="Start_Command"
Defines the command to start the port.
**IPPORT**

**STOP**="Stop_Command"

Defines the command to stop the port.

**EXIT 13**="User_Exit"

User code to be run before the STOP command and before the ACTIVATE command. If the user exit return code is nonzero, the STOP or ACTIVATE command will not be issued.

**DESC**="User_Text"

The user defined text.

**ACTMON**=actmon_Def

Defines active monitoring for groups of resources in the network. For additional information see [ACTMON](#) on page 209.

**FORMAT**=PORT

Specifies that port data will be collected to check the status of the resource. PORT is the only valid option for IPPORTs.

**STATUS**=(status1,.....,statusn)

Defines the expected (AON) status of the resource.

**Example:**

The following example defines the SMTP port, socket number 25, to be monitored every 5 minutes using SNMP GET request based on the IPPORT (restype) policy for all ports. A NORMAL status comes from the IP390 level.

```
IPPORT SMTP,
    SP=NMPIPL10,
    PORT=25,
    TCPNAME=SMTP32,
    ACTMON=IPPORT,
    DESC='SMTP Socket 25'
```

**Example:**

The following example defines the NetView web browser socket to be monitored by default every 10 minutes if the ACTMON IPPORT statement does not specify the INTVL keyword and its value:

```
IPPORT DSIWBTSK,SP=NMPIPL10,
    PORT=8008,
    TCPNAME=E330WTCP,
    CMDTYPE=NETV,
    STATUS=NORMAL,
    ACTIVATE="START TASK=DSIWBTSK",
    DESC= "NetView Web Browser Socket"
```

If the socket monitoring detects that the socket is down, automation issues a NetView START TASK=DSIWBTSK command. DSIWBTSK will also be available for the session status functions since SESSTAT=YES, by default.
The IPROUTER control file entry defines IP routers to IP390 and optionally defines active monitoring of the router.

```
|IPROUTER| AON_Name, SP=stack_name, OPTION=IP390 |
|IPADDR=ipaddr, HOSTNAME=host_name, INTVL=interval |
|ACTMON=Actmon_Def, FORMAT=PING, SNMP |
|STATUS=(status1, ..., statusn) |
|MIBVAR1, ..., MIBVARn=(mib_var, operator, value) |
|CORRELATE N, EXIT14='FKXEXT14 ...
|EXIT15='FKXEXT15 ...
```

**AON_Name**
The unique name associated with the router that is used by AON/TCP. The name must be a valid REXX symbol.

**Stack_Name**
The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

**OPTION=IP390**
The only valid entry is IP390.

**IPADDR**
Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 256.256.256.256 format. Alphabetical characters are not valid.

**HOSTNAME**
Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The host_name variable is case-sensitive. The following is an example of a host name:

```
mrpres.whitehouse.capital.usa
```
INTERVAL
Defines the monitoring interval in hh:mm format. This is required for proactive monitoring. If the interval is defined for IPSTAT but not specified monitoring will not occur.

ACTMON_Def
Defines active monitoring for groups of resources in the network. For additional information see "ACTMON" on page 209.

FORMAT
Specifies one of the following options to be used to determine the resource status:

PING Pings the resource to check its status.

SNMP Uses SNMP MIB polling to check the status of the resource.

STATUS
Defines the expected (AON) status of the resource.

MIBVAR1 – n
Multiple MIBVAR statements can be defined and is needed only for SNMP thresholding (requires FORMAT=SNMP).

CORRELATE Y|N
Set to Y if trap correlation of IPHOST’s/IPROUTER’s/IPTN3270’s and their respective interface is desired. The CORRELATE parameter should only be set to Y for resources when SNMP is available. All known interfaces will be used to correlate the status. Correlation is only recommended for interfaces installed on critical IPHOST’s or IPROUTER’s. N is the default.

EXIT14
Defines additional processing of SNMP interface table. The default value is NONE. Invoked for FORMAT=SNMP.

EXIT15
Defines additional processing of user defined thresholds (MIBVAR). The default value is NONE.

Example:
The following example defines ROUTER1 as part of the NYROUTERS policy grouping to be monitored every 15 minutes through SNMP.

ACTMON IP390,OPTION=IP390,INTVL=01:,STATUS=NORMAL
ACTMON IPROUTER,OPTION=IP390,INTVL=00:30,STATUS=NORMAL
ACTMON NYROUTERS,OPTION=IP390,INTVL=00:15,
FORMAT=SNMP

IPROUTER ROUTER1,
SP=NMP1PL10,
OPTION=IP390,
IPADDR=9.67.2.299,
HOSTNAME=router1.plant.floor.co
ACTMON=NYROUTERS
## IPTN3270

The IPTN3270 control file entry defines TN3270 servers to IP390 (used for session management) and optionally defines active monitoring for the TN3270 server.

```plaintext
IPTN3270: name, IPADDR=address, SP=stack_name, TNPORT=portnum, SYSTEM=TN3270, DATACOL=module, DROP=module, FORMAT=SNMP, HOSTNAME=name
```

- **name**: The unique name associated with the TCP/IP host that is used by AON/TCP. The name must be a valid REXX symbol.
- **IPADDR=address**: Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 255.255.255.255 format. Alphabetical characters are not valid.
- **SP=stack_name**: The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.
- **TNPORT**: The TN3270 port number.
- **SYSTEM**: The type of service point this server resource is.
- **DATACOL**: Tells FKXE2COL which SNMP MIBs to query to get the TN3270 session data. This is required for any TN3270 system.
  
  **Note**: For Cisco CIP, use FKXEXCIP.
- **DROP**: Tells FKXE2210 which routine to use to process session drop requests.
- **FORMAT**: Specifies that one of the following options to be used to determine resource status:
  - **PING**: Pings the resource to check its status.
  - **SNMP**: Uses SNMP MIB polling to check the status of the resource.
HOSTNAME
Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. Hostname is case-sensitive. The following is an example of a hostname:

```
mrpres.whitehouse.capital.usa
```

TNIPADDR=address:yy
Dotted decimal address for the TN3270 server. Where:
- **x** is a single digit integer from 0 to 9
- **yy** is the port number of the telnet server. The default is 23.

CORRELATE Y|N
Set to Y if trap correlation of IPHOST’s/IPROUTER’s/IPTN3270’s and their respective interface is desired. The CORRELATE parameter should only be set to Y for resources when SNMP is available. All known interfaces will be used to correlate the status. Correlation is only recommended for interfaces installed on critical IPHOST’s or IPROUTER’s. N is the default.

INTVL
Defines the monitoring interval in hh:mm format. This is required for proactive monitoring.

ACTMON
Defines active monitoring for groups of resources in the network. For additional information, see "ACTMON" on page 209.

STATUS
Defines the expected (AON) status of the resource.

MIBVAR1 – n
Multiple MIBVAR statements can be defined and is needed only for SNMP thresholding ( requires FORMAT=SNMP).

EXIT14
Defines additional processing of SNMP interface table. The default value is NONE. Invoked for FORMAT=SNMP.

EXIT15
Defines additional processing of user defined thresholds (MIBVAR). The default value is NONE.

Examples:
The following is an example of IPTN3270:

```
IPTN3270 IBM2210,
  SP=MVSA,
  IPADD=99.200.99.100,
  TNPORT=23,SYSTEM=TN3270,
  DATACOL=FKXEX216,
  INTVL=15,
  FORMAT=SNMP,
  STATUS=(NORMAL,DEGR*,THRESH*),
  HOSTNAME=IBM2210.rtp.lab.tivoli.com
```

The following is an example of IPTN3270 for Cisco CIP.

```
IPTN3270 CIP3270,
  SP=TVT2017,
  IPADD=99.200.10.100,
  TNIPADD=99.200.10.101:23,
  TNIPADD2=99.200.10.102:23,
  TNIPADD3=99.200.10.103:623,
```
The LSTHRESH (large scale thresholding) control file entry counts network-wide events of a particular type and sets a threshold of the number of times the event can occur over a specific period of time.

**LSTHRESH**

```
LSTHRESH~NAMESERV,~COUNT=\(nn, hh:mm\),~EVENTLIM=nnn,~NOTIFY=ALL,THR
```

**NAMESERV**
Nameserver failure threshold for AON/TCP.

**COUNT**
Specifies the number of events (1–999) and the time interval. If the specified number of events is exceeded within the specified interval, AON notifies operators and initiates other actions, depending on the type of threshold and the type of automation.

**EVENTLIM**
The maximum number of total events that AON tracks (only the date and time stamp are saved) for threshold analysis. This number should be significantly higher than COUNT. The dates and time stamps are compressed into 255-byte global variables, so that 25 events can be stored in each variable. For example, if EVENTLIM=200, AON uses eight global variables to track up to 200 events at any one time.

**NOTIFY**
This optional parameter is used only with AON/TCP. NOTIFY specifies whether operators are notified on each occurrence of the event, or only when the threshold is exceeded. ALL specifies that operators are notified on each occurrence. THR specifies that operators are notified only when the threshold is exceeded. ALL is the default.

**Example:**

```
LSTHRESH NAMESERV,COUNT=(10,00:10),EVENTLIM=50,NTFY=THR
```

In this example, Name Server alerts are tracked by AON/TCP. A maximum of 50 events are saved in global variables before wrapping occurs.

If 10 Name Server alerts are received within a 10 minute time period AON/TCP notifies appropriate personnel that the name server threshold was exceeded. This signifies a possible network-wide performance problem.
The MONIT control file entries control reactivation intervals for failing network resources. AON attempts to reactivate failed resources and, optionally, notify operators at specified intervals. You can include multiple MONIT entries in the control file, one for each resource, one for the resource type, and another for the default actions. At least one MONIT control file entry is required; you must code a MONIT DEFAULTS entry.

If your installation has special requirements for reminder interval automation processing, define AON user exits for specific MONIT intervals. These are specified in MONIT entries or in the ENVIRON EXIT entry.

**MONIT**

<table>
<thead>
<tr>
<th>MONIT</th>
<th>DEFAULTS</th>
<th>INTVL=(*,hh:mm,Y),EXIT09= User Exit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ResType</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ResName</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Res*</td>
<td></td>
</tr>
</tbody>
</table>

**DEFAULTS**
System wide defaults. A MONIT DEFAULTS entry is required.

*ResType*
Type of resource to automate. AON recognizes three resource types: subsystems, MVS components, and network components. Examples of networking component names are LU, PU, LINE, CDRM, NCP, APPL, and LINKSTA. The following wildcard characters can be used:
- `*` Multiple character wildcard
- `%` Single character wildcard

For example, both PU0* and P%%1 match for PU01.

*ResName*
The name of the resource

*Res*
A group of Resource levels

**INTVL**
Defines the MONIT intervals in `hh:mm` format. Values are in the range of 00:01 – 24:00. Intervals are cumulative. INTVL are time delays from the previous activation. In addition to the `time`, INTVL defines monitoring intervals and notification action(s) to take at each interval. There is no default value.

- **Y** Invokes the Notification Policy defined for the resource or event type.
- **N** Does not invoke any Notification Policy. Sends messages to the notification operators.
- **YF** Notify operators based on your NOMOMONS Notification Policy and repeat recovery monitoring at the last interval specified until the resource recovers or the control file is reloaded.
- **NF** Repeat recovery monitoring at the last interval specified, but do not invoke Notification Policy.
EXIT09

EXIT09 is the EZLECATV user exit. This user exit allows override of any of the MONIT values, for example, you can implement a NOTIFY Policy for each interval.

Usage Note:

AON uses the most specific MONIT entry it finds to guide monitoring and recovery activity. If DEFAULTS, PU, and PU02 MONIT statements are defined, and PU01 fails, AON uses the MONIT statement for the resource type PU. If PU02 fails, AON uses the MONIT statement for the resource named PU02.

Examples:

The following MONIT entry example causes AON to attempt to reactivate PUs 10 minutes after a failure and again 20 minutes after the first reactivation attempt. The final recovery attempt is made two hours after the initial failure. Operator notifications are issued for all attempts.

MONIT PU,INTVL=(00:10,Y),INTVL=(00:20,Y),INTVL=(01:30,Y)

The following MONIT entry example attempts to reactivate resource TA1T1000 five minutes after a failure, again one hour after the first reactivation attempt, and continue to repeat the attempts at 8-hour intervals. Operator notifications are issued at five minutes, the third hour, and subsequent 8-hour intervals until the resource becomes active, the timer is purged, or an operator inactivates the resource.

MONIT TA1T100,INTVL=(00:05,Y),INTVL=(01:00,N),INTVL=(08:00,N),
INTVL=(08:00,N),INTVL=(08:00,YF)

MONITOR

The MONITOR control file entry defines monitoring of gateway connections. The MONITOR entry permits the recovery of gateways if status changes occur. One entry for each distributed host is allowed in the control file at the focal point host. One entry for each focal point and backup focal point is valid at the distributed host.

MONITOR domain—,INTERVAL=hh:mm—,TIMEOUT=ss—

domain

Identifies the target host domain.

INTERVAL

Specifies the interval of time used to check that an NNT session exists for the gateway operator to the domain.

TIMEOUT

Specifies in seconds how long AON waits for a response from a specified domain. If the TIMEOUT value is exceeded, AON initiates recovery procedures for the gateway NNT session by reinitializing the logon process.

Usage Notes:
MONITOR

One entry is required for each gateway to be monitored. On the monitor cycle, a query is sent to the distributed host with a TIMEOUT value. If the query TIMEOUT value is exceeded, automation then starts procedures to re-establish the NNT session.

Example:

In this example, AON determines that an NNT session exists for a gateway operator to domain CNM01 every five minutes. If AON does not receive a response from CNM01 within 30 seconds, it initiates recovery procedures for the gateway NNT session. If the NNT is not active, messages are issued to indicate the status of the NNT session. EZLE1IMN is initiated for the gateway operator.

```
MONITOR CNM01, INTERVAL=00:05, TIMEOUT=30
```
ncpname
Identifies the name of the NCP (PU name). You can use wildcard characters to address version changes reflected by NCP naming conventions without updating the control file for each NCP generation cycle. However, each physical NCP must have an entry to maintain the correct relationship between the NCP name and the linkstation used to load the NCP.

HOST
Identifies the domain ID of the recovery host. The recovery host is the channel attached host which replies to the dump and reloads outstanding replies. Other channel attached hosts respond no (N) to the dump or reloads an outstanding reply if AON/SNA is installed and an NCPRECOV statement is coded for the NCP with this host name.

DUMP
Specifies dump responses for the recovery host. The first parameter is the noncritical dump response. The second parameter is the critical dump response that is issued when the threshold is exceeded.

RELOAD
Specifies reload responses for the recovery host. The first parameter is the noncritical reload response. The second parameter is the critical reload response, which is issued when the threshold is exceeded. Define the value for each of these parameters as either Y (Yes) or N (No).

DUMPTIME
Specifies the amount of time (hh:mm) a dump of this NCP should require, after which an operator is notified that excessive time has been used for dumping this NCP. The time is measured as that between when the response to outstanding reply is issued and when the dump complete message is issued.

LOADTIME
Specifies the amount of time (hh:mm) a load of this NCP should require, after which an operator is notified that excessive time has been used for loading this NCP. The time is measured as that between when the response to outstanding reply is issued and when the load complete message is issued.

LOADMOD
The NCP load module name, used to load an NCP. The default value is the NCP PU name, as specified in the NCP definition.

LINKSTA
Specifies the link station used to load the NCP and creates a reply to the option that is to be loaded. This must be accurately defined for the NCP. It is used for early detection of NCP failures when linkstation failures are received.

DUMPSTA
Specifies the link station used to dump the NCP. Used in reply to the option to dump. This is an optional parameter.

EXIT01
Specifies a user-defined command or command list. The program defined in
NCPRECOV (SNA)

The EXIT01 exit runs after the dump message receives a response. Sample exit routines are provided (FKVEX01 in CNMSAMP). This is an optional parameter.

EXIT02

Specifies a user-defined command or command list. The program defined in the EXIT02 exit runs after the dump is complete and the dump time exceeded timer is purged. Sample exit routines are provided (FKVEX02 in CNMSAMP). This is an optional parameter.

AUTOINIT

Specify Y (YES) or N (NO). Y is the default. Specifies whether the NCP should be activated at startup of AON if it is not already active.

EXIT03

Specifies a user-defined command or command list. The program defined in the EXIT03 exit runs after the load message is responded to. Sample exit routines are provided (FKVEX03 in CNMSAMP). This is an optional parameter.

EXIT04

Specifies a user-defined command or command list. The program defined in the EXIT04 exit runs after the load is complete and the load time exceeded timer is purged. Sample exit routines are provided (FKVEX04 in CNMSAMP). This is an optional parameter.

Usage:

• If you want to use one of the sample exits, copy them from the sample library into your DSICLD library.
• After issuing the message IST530I (GBIND PENDING) failure on the NCP and after reloading, the non-recovery host activates the NCP.
• Define an NCPRECOV statement to recover each NCP.
• Coding AUTOINIT=Y (or using it as the default) drives AON recovery processes if the NCP is not active during AON initialization. Therefore, you may want to code AUTOINIT=N for any less critical NCP if you have a large network.

Example:

In this example, assume the NCP is defined as NCPZZ, and a LOAD is requested for domain CNM01:

NCPRECOV NCPZZ,HOST=CNM01, DUMP=(Y,N), RELOAD=(Y,N), LINKSTA=123-S, DUMPSTA=630-S, LOADTIME=00:15, DUMPTIME=00:10, LOADMOD=NCPYY, AUTOINIT=N

If the critical threshold has not been exceeded, the response for a dump request is yes (Y). If the critical threshold is exceeded the response for a dump request is no (N). Link station 630-S is to be used for both the load and the dump of the NCP. The load of the NCP is expected to take no longer than 15 minutes. A dump of the NCP is expected to complete within 10 minutes. No exits are used. If the dump definition is on CNM01 and CNM02, and the dump or load request is displayed on CNM02, the response on CNM02 is no (N) under all circumstances.
The NOTIFY control file entry defines notification actions for the NetView operators who receive AON notifications. These notifications can be messages, alerts, DDF updates, and Inform Policy actions; such as, beeper or e-mail.

```
--- NOTIFY --- DEFAULTS ---
  \ ResType \ ResName \ Res* \ Event_Type \ ALERT
  NO \ NO \ NO \ YES \ TEC

--- INFORM ---
  \ Inform_Policy_Name \ MSG \ DDF
  NO \ NO \ NO

--- EXITIO ---
  \ 'User_Exit'
```

**DEFAULTS**
Notification Policy at a system wide, or defaults, level

**ResType**
Notification Policy based on a type of resource

**ResName**
Notification Policy based on a particular resource name

**Res**
Notification Policy based on a range of resource names

**Event_Type**
Event_Type can be one of the following:

- **CRITTHRS**
  Critical Automation Threshold Exceeded

- **NOMOMONS**
  No More Monitoring Intervals Defined

- **REMINd**
  Reminder that a resource is still down

- **NAMESERV**
  NameServer Failure Threshold exceeded

**ALERT**
Specifies when to generate MSU notifications. MSUs are required for Tivoli Enterprise Console® events.

- **YES**
  Specifies generate an MSU.

- **NO**
  Specifies do not generate an MSU. No is also the default.

- **TEC**
  Specifies to generate an MSU so that the Tivoli Enterprise Console will receive the notification.

**INFORM**
Specifies the use of pager/beeper or e-mail that is defined in the CONTACT
NOTIFY

deploy statement of the specified Inform Policy. Inform actions take place when
the notify and event-type criteria are met and an Inform Policy name is
supplied.

MSG
Specifies whether to generate a message to NTFYOPs.
YES  Specifies display a message.
NO   Specifies do not display a message. No is the default.

DDF
Specifies how the DDF component is updated.
YES  Log event to DDF.
NO   Do not log events to DDF. No is the default.

EXIT10
Is a user exit that enables the override of any of the notification actions.

Example:

This example defines DEFAULTS Notification Policy as Tivoli Enterprise Console
updates, and DDF updates for all resources. Messages are not passed:
NOTIFY DEFAULTS,ALERT=TEC,MSG=NO,DDF=YES

Tivoli Enterprise Console updates occur and DDF records are created.

Example:

This example defines Notification Policy for an event type of CRITTHRS, Critical
Threshold Exceeded. In this case, an INFORM Policy of Critical is invoked to
e-mail support personnel. If MSG=YES is coded then messages are generated.
NOTIFY CRITTHRS,
INFORM=Critical,
MSG=YES

NTFYOP

The NTFYOP control file entry identifies NetView operators who receive AON
notifications. These notifications are AON messages sent to the operator’s
workstation and can be based on job responsibilities, such as all IP nameservers.
The messages can be held until the operator clears them. You should assign
operators responsible for system operation to receive all AON messages or a subset
of those messages. NTFYOP is an optional control file entry, but you should
customize these entries.

```
| NTFYOP operid,OPER=desc,CLASS=(class,class,class)
| ,NTFY=Y,HELDMSG=(type,type,type)
```

264 Administration Reference
**operid**

Identifies the NetView operators to receive notification messages. This name cannot exceed eight characters, and should be defined in the NetView DSIOPF member. The reserved name SYSOP does not need a DSIOPF entry. Specifying SYSOP controls which messages are sent to the system console, and whether those messages are held or rolled off the screen.

**OPER**

Associates an optional description with the specified operator ID. This description may be up to 20-characters long and must be enclosed in quotes if it contains blanks.

**CLASS**

Specifies the message notification classes to be sent to this operator ID. Up to 10 message classes may be specified. There are no default message classes. The class assigned to the notify operators may be changed using the SETNTFY command. The following table lists the AON message classes:

<table>
<thead>
<tr>
<th>Message Class</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Errors</td>
</tr>
<tr>
<td>01</td>
<td>Down</td>
</tr>
<tr>
<td>02</td>
<td>Up</td>
</tr>
<tr>
<td>03</td>
<td>Degraded/Performance</td>
</tr>
<tr>
<td>04</td>
<td>Manual Intervention Required</td>
</tr>
<tr>
<td>07</td>
<td>Critical Threshold Exceeded</td>
</tr>
<tr>
<td>08</td>
<td>Frequent Threshold Exceeded</td>
</tr>
<tr>
<td>09</td>
<td>Infrequent Threshold Exceeded</td>
</tr>
<tr>
<td>10</td>
<td>SNA Automation</td>
</tr>
<tr>
<td>11</td>
<td>VTAM Detected Storage Problems</td>
</tr>
<tr>
<td>12</td>
<td>X.25 Automation</td>
</tr>
<tr>
<td>15</td>
<td>SNBU Automation</td>
</tr>
<tr>
<td>20</td>
<td>VTAM Subarea Automation</td>
</tr>
<tr>
<td>21</td>
<td>NCP, Linksta</td>
</tr>
<tr>
<td>22</td>
<td>Line</td>
</tr>
<tr>
<td>23</td>
<td>PU</td>
</tr>
<tr>
<td>24</td>
<td>CDRM</td>
</tr>
<tr>
<td>25</td>
<td>CDRSC</td>
</tr>
<tr>
<td>26</td>
<td>APPL</td>
</tr>
<tr>
<td>27</td>
<td>Session</td>
</tr>
<tr>
<td>40</td>
<td>Messages to forward to AOC</td>
</tr>
<tr>
<td>50</td>
<td>APPN Automation</td>
</tr>
<tr>
<td>60</td>
<td>TCP/IP Automation</td>
</tr>
<tr>
<td>61</td>
<td>IP Router</td>
</tr>
<tr>
<td>62</td>
<td>Nameserver</td>
</tr>
<tr>
<td>63</td>
<td>IP Socket</td>
</tr>
<tr>
<td>64</td>
<td>MVS TCP/IP Automation</td>
</tr>
</tbody>
</table>
Table 16. AON Message Classes (continued)

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>Tivoli NetView for UNIX automation</td>
</tr>
<tr>
<td>66</td>
<td>Service Point</td>
</tr>
<tr>
<td>67</td>
<td>IP Host</td>
</tr>
<tr>
<td>68</td>
<td>Interface</td>
</tr>
<tr>
<td>69</td>
<td>IP Link</td>
</tr>
<tr>
<td>90</td>
<td>AON Automation</td>
</tr>
</tbody>
</table>

**NTFY**
Specifies whether the operator receives notification messages. The default is YES.

**HELDMSG**
Specifies the type or types of messages that should be held on the operator’s screen. The first character of the HELDMSG type can specify the desired message type. Multiple types can be specified. Following are the message types:
- **I** or INFO
  - Informational messages
- **W** or WARN
  - Warning messages
- **E** or ERROR
  - Error messages
- **A** or ACTION
  - Action messages

**Usage Notes:**
The NTFY=Y|N can be used to temporarily turn on and off receiving notification messages. The NTFY parameter can be changed while NetView is up by using the SETNTFY command.

In a distributed environment, the notify operators are defined at the focal point host. In a single NetView host environment, the host is considered the focal point host.

When a notification message is received, and none of the NTFYOP operators are signed on to NetView, the message is sent to the system console. Additionally, if HELDMSG is specified on the SYSOP entry, those message types specified are held on the console screen. Operator intervention is required to clear the messages from the console screen.

If HELDMSG is not specified on NTFYOP entries or the ENVIRON SETUP entry, the default is to roll all messages that go to the system console. NTFYOP HELDMSG parameters override the defaults specified in the ENVIRON SETUP entry. For example, specifying HELDMSG=(W,I) on the ENVIRON SETUP entry and HELDMSG=E on the NTFYOP entry results in only error messages being held for that operator. However, if HELDMSG is not specified on the NTFYOP entry, the ENVIRON SETUP default is used.

This example defines an operator to receive AON notifications only:
```
NTFYOP NETOP1,OPER='NETOPER 1',CLASS=(10)
```
In the previous example, NETOP1 is defined to receive AON notifications. The notify parameter was not coded and defaults to NTFY=Y.

This entry is a valid example for all operating systems in which NetView is installed.

Example:

This example specifies that the system console receives only class 11 and 12 messages and that the error messages are frozen:
NTFYOP SYSOP,OPER='SYSOP',CLASS=(11,12),NTFY=Y,HELDMSG=E

In the previous example, SYSOP is defined to receive notifications from only AON for only message classes 11 and 12. SYSOP does not have to be defined in DSIOPF. Class 11 and 12 messages are sent to the system console and only error messages are held.

**NV6000 (TCP/IP)**

The NV6000 entry defines Tivoli NetView service points to AON/TCP, enabling AON/TCP to send commands to those service points.

```
NV6000
   DEFAULTS
   NetView_for_AIX_servicepoint, SP=NVAIX_puname, APPLID=sp_id
   HIER2=hier2_appl
   HIER3=hier3_device
   FAILURE=(restype1, restype2, restype3), FILTER=Y
   PINGCNT=pingnum
   PINGTIME=pingtime
   IPADDR=ipaddr
   HOSTNAME=hostname
```

The `NetView_for_AIX_servicepoint` name must be the same name that is defined on the Tivoli NetView service point. When Tivoli NetView sends an alert to NetView, this name is forwarded in the first alert hierarchy position with a resource type of SP. You must define this name in the SNA SERVICES portion of the Tivoli NetView definitions. You can verify this name by examining the Tivoli NetView alerts coming from the service point you are defining and looking at the first element of the resource hierarchy.

**SP** Identifies the systems network architecture (SNA) PU name used to route run commands (RUNCMD) to the Tivoli NetView host. You must also define this name in the SNA SERVICES portion on the Tivoli NetView service point. If you have already set up the Tivoli NetView service point, you can get the value from the SNA SERVICES portion and copy it.

**APPLID** Identifies the application ID of the process running on Tivoli NetView that receives and processes run commands (RUNCMD) from the NetView session on the host. This is defined in options of the SPAPPLD and TRALERTD processes for Tivoli NetView.
Hier2

Identifies the value of the field in the second alert hierarchy level with a resource type of TP of the alerts that Tivoli NetView sends to NetView. Defining this value ensures that alerts from AON are filed with Tivoli NetView alerts in the Tivoli NetView hardware monitor and NetView management console.

Hier3

Identifies the value of the field in the third alert hierarchy level with a resource type of DEV of the alerts that Tivoli NetView sends to NetView. Defining this value ensures that alerts from AON are filed with Tivoli NetView alerts in the Tivoli NetView hardware monitor and NetView management console.

Failure

Indicates which Tivoli NetView alerts AON/TCP should process for outage detection, where restype is HOST, LINK or INFC. You can define up to three resource types in any combination you need. If the alert names a host as the failing device, the resource type is HOST (hostname is in dotted alphanumeric format with at least one alphabetic character). Generally only the first word of the host name (to the first dot) is provided by the alert hierarchy as the resource name.

Depending on the alert, the entire host name is usually available elsewhere in the alert. If the alert names an IP address as the failing device, the resource type is INFC (IPADDR is in dotted decimal format). Links are connections between two TCP/IP devices, especially on routers and gateways. Alerts for link availability sometimes accompany one of the other alerts, as well. Because at least two alerts are received for each failure and recovery (HOST and INFC), this can reduce processing for outages to one iteration instead of two. There is still some overhead associated with failure processing.

This value is checked in the automation table, so it provides an effective way to reduce overhead due to multiple failure processing for a single failure event on a single resource.

Filter

Indicates whether only resources with definitions in the control file entries should be automated and monitored from the NetView automation table. If FILTER=N, all alerts from Tivoli NetView are processed and all outages are monitored. If FILTER=Y, only alerts for those devices predefined in the control file in TCPIP statements are processed and monitored for outages.

Pingcnt

Indicates the number of pings (1–99) to be run on the Tivoli NetView service point with the ping command whenever a node is checked for availability and performance. You should perform at least two pings to ensure that the first failure is not due to node search time.

Pingtime

Indicates, in seconds, how long to wait for each ping sent by the Tivoli NetView service point to the node.

Ipaddr

Consists of dotted decimal IP addresses up to 17 characters including dots. They must be in the 256.256.256.256 format. Alphabetical characters are not valid.
Because some hosts have multiple IP addresses, you can define a list of IP
addresses. If a failure for any of these IP addresses is received, failure
processing is done for the host. Because AON/TCP uses the first IP
address in the list as a target for doing pings, put the most important or
frequently used IP address first in the list.

**HOSTNAME**
Indicates the fully-qualified TCP/IP host name, using up to 30 characters
including dots. *Hostname* is case-sensitive. The following is an example of a
hostname:

```
mrpres.whitehouse.capital.usa
```

**Note:** If you are going to define multiple Tivoli NetView service points, you can
define a NV6000 DEFAULTS entry for parameters that apply to all Tivoli
NetView service points. Valid keywords on a DEFAULTS statement are
FAILURE, FILTER, PINGCNT, and PINGTIME.

---

**RECOVERY**

The RECOVERY control file entry provides an ON/OFF switch to control recovery
of a network resource. This switch can be turned on or off by specific resource
name, by resource type, or for the entire network. If the RECOVERY entry
indicates that automation is turned off, those actions and responses are not
automated.

Multiple RECOVERY entries can be coded in the automation control file, one for
each resource, one per resource type, or one for the default actions. Any
combination of RECOVERY flags can be coded, and the most specific RECOVERY
flag that is found controls whether automation is on or off.

The RECOVERY entry is required. A RECOVERY DEFAULTS entry must be
specified.

AON is designed to meet most recovery processing needs. However, your
installation may have special requirements that require you to tailor recovery
automation processing. You can code AON user exits for specific recovery
processing. User exits for recovery processing are specified on the RECOVERY
statement or on the ENVIRON EXIT statement.

---

**RECOVERY**

```
RECOVERY
  DEFAULTS
  | res_name
  | res_type
  | Day_Policy
  | AUTO
  | ,NOAUTO=(day,x,start,end)
  | ,MSG
  | ,NOTIFY=intv
  | ,CHECK=intv
```
RECOVERY

[command parms]

,EXIT07=(command parms)

,MSGCLASS=mm

DEFAULTS
Used when the resource name and resource type are not specified. This is the least specific RECOVERY entry that can be specified. A RECOVERY DEFAULTS entry is required.

res_name
The name of the resource to automate.

res_type
The resource type to automate. Examples of SNA networking component types are LU, PU, LINE, CDRM, NCP, APPL, and LINKSTA. You can use the following wildcard characters for res_name and res_type:
* Multiple character wildcard
% Single character wildcard

For example, both PU0* and P%%1 are a match for PU01.

Using wildcard characters in the res_type field is not valid with AUTO=YA and AUTO=NA.

Day_Policy
The special day specification:
SUN or SUNDAY
MON or MONDAY
TUE or TUESDAY
WED or WEDNESDAY
THU or THURSDAY
FRI or FRIDAY
SAT or SATURDAY
Month/date_spec/year (examples: JAN/1/2000 or JAN/LAST-30/2000)
Month/day_spec/year (examples: JAN/SAT/1ST/2000 or JAN/SAT/LAST-4/2000) In this case SAT cannot be SATURDAY - only three characters are allowed.
Special_day_name (example: My_Birthday or HOLIDAY or NEW_YEARS_DAY)

AUTO
Defines whether automation is valid.
Y If specified as AUTO=Y, recovery is on. AUTO=Y is the default setting.
N If specified as AUTO=N, recovery is off.
YA If specified as AUTO=YA, recovery is on for this resource and its lower nodes. The lower nodes for a particular resource are determined by issuing the VTAM command:
D NET, ID=nodeid,E

The nodeid parameter represents the node ID of the higher node.

AUTO=YA is valid only with SNA resources. AUTO=YA is not valid when the res_type field contains wildcard characters.
If specified as AUTO=NA, recovery is off for this resource and its lower nodes. The lower nodes for a particular resource are determined by issuing the VTAM command:

```
D NET,ID=nodeid,E
```

The nodeid parameter represents the node ID of this resource.

AUTO=NA is valid only with SNA resources. AUTO=NA is not valid when the res_type field contains wildcard characters.

**Note:** At initialization, RECOVERY resource, AUTO=YA or AUTO=NA builds a list of RECOVERY AUTO=Y or AUTO=N statements in storage for the specified resource and all of its displayable subordinates. If this parameter is changed, a reload of the control file does not drive the process. Either the EZLEHRCY program can be run or the SETAUTO command must be used. If you use YA or NA extensively, reestablishing all hierarchies through EZLEHRCY execution can cause a significant amount of overhead.

**NOAUTO**

Defines specific times when RECOVERY is off, where:

- **dayx**
  - Can be expressed as:
  - An * (asterisk) to indicate every day
  - SUN or SUNDAY
  - MON or MONDAY
  - TUE or TUESDAY
  - WED or WEDNESDAY
  - THU or THURSDAY
  - FRI or FRIDAY
  - SAT or SATURDAY
  - Month/date_spec/year (examples: JAN/1/2000 or JAN/LAST-30/2000)
  - Month/day_spec/year (examples: JAN/SAT/1ST/2000 or JAN/SAT/LAST-4/2000) In this case SAT cannot be SATURDAY - only three characters are allowed.
  - Special_day_name (example: My_Birthday or HOLIDAY or NEW_YEARS_DAY)

- **start**
  - Defines the starting time when RECOVERY is off. The time must be specified using the 24-hour clock in the **hh:mm** format. Valid values are in the range of 00:00 to 23:59.

- **end**
  - Defines the ending time (the time when RECOVERY is back on). The time must be specified using the 24-hour clock in the **hh:mm** format. Valid values are in the range of 00:00 to 23:59. The ending time cannot be less than the starting time, unless the day-of-week is coded as an asterisk (*)

**MSG**

Specifies whether availability messages (EZL504I and EZL509I) for failures and recoveries are sent. This includes updating DDF and log files.

**Notes:**

1. If AUTO=N and MSG=N, no messages about this resource appear in AON logs or DDF.
RECOVERY

2. If AUTO=Y and MSG=N, operators still receive messages concerning
   recovery monitoring, active monitoring, and thresholding.

3. The AON reports facility depends on availability messages for accurate
   reporting. You may not get the correct resource type associated with the
   resource if MSG=N. This applies to automated and non-automated
   resources.

NOTIFY

Defines the interval that AON uses to notify the operators if the
APPLICATION is not active. This option is only used for APPL resource types
in AON.

\textit{intv}

Defines the time interval AON uses to notify the operators that the
application is still not operational. You must specify the time using the
24-hour clock in the \textit{hh:mm} format. Valid values are in the range of 00:00 –
23:59.

CHECK

Defines the interval to check the status of the application. This option is used
only for APPL resource types in AON.

\textit{intv}

Defines the time interval to be used to check on the application to
determine the status of the application. The time must be specified using
the 24-hour clock in the \textit{hh:mm} format. Values can range from 00:00 to
23:59.

EXIT07

Overrides exit information specified in the ENVIRON EXIT control file entry.

MSGCLASS

The message class that resources define with this recovery statement is used
for messages issued on their behalf. This is an addition to the message class
already defined for the message by AON based on component, resource type,
or status.

Usage:

Only one RECOVERY entry is used to determine whether automation is on or off.
The first RECOVERY entry found is the one that is used. The routine searches in
the following order:
1. Resource name
2. Resource type
3. DEFAULTS

Note: The DEFAULTS NCP entry is required.

The RECOVERY entry can be added, changed, or deleted dynamically using the
SETAUTO or TIMEAUTO commands. The EZLECAUT common routine checks the
RECOVERY entry definition and evaluates whether a resource is to be
automatically recovered.

The following example defines a sample DEFAULTS entry. In this example,
recovery activity and all resources are automated.

RECOVERY DEFAULTS,AUTO=Y

Example:
The following example defines an entry where RECOVERY is off every day for a specific time period. In this example, RECOVERY is off daily between 10 p.m. and 02 a.m.

RECOVERY DEFAULTS,AUTO=Y,
NOAUTO=(*,22:00,02:00)

Example:

The following example defines an entry where automation is off Sunday evening through Monday morning. In this example, RECOVERY is off from 6 pm on Sunday until 2 am on Monday. Two entries were defined, one for Sunday evening through midnight (23:59); the other for Monday morning.

RECOVERY DEFAULTS,AUTO=Y,
NOAUTO=(SUNDAY,18:00,23:59),
NOAUTO=(MONDAY,00:00,02:00)

Example:

The following example defines an entry where RECOVERY is off for SNA applications, but is on for all other VTAM resources.

RECOVERY DEFAULTS,AUTO=Y
RECOVERY APPL,AUTO=N

Example:

The following example defines an entry where RECOVERY is off for SNA applications and all SNA networking lines, but is on for all other network resources.

RECOVERY DEFAULTS,AUTO=Y
RECOVERY APPL,AUTO=N
RECOVERY LINE,AUTO=N

Example:

The following example defines an entry where RECOVERY is off for all SNA applications except the one named CICS1. The NOTIFY and CHECK parameters must be specified when AUTO=Y for an SNA application is specified.

RECOVERY APPL,AUTO=N
RECOVERY CICS1,AUTO=Y,NOTIFY=01:00,CHECK=00:15
RECOVERY MSM*,AUTO=Y

Use of RECOVERY Policy example:

The following example defines an entry where RECOVERY is enabled for all resources on a calendar day called HOLIDAY except between the hours of midnight and 6 A.M.. In this example, HOLIDAY would be defined in the DSISCHED file.

RECOVERY HOLIDAY,AUTO=Y,NOAUTO=(*,00:00,06:00)

The following example defines a RECOVERY policy where automation is enabled for all resources whose names begin with ABC except between Midnight an Noon on the last Saturday of every month.

RECOVERY ABC*,AUTO=Y,NOAUTO=(*/SAT/LAST/*/00:00,12:00)
The RESIDENT control file entries in the control file define programs that are loaded into main storage to improve performance.

```
RESIDENT program
```

*program*

Specifies the filename of the program or the command name to be loaded.

**Usage Notes:**

The programs defined with this entry are loaded into main storage during initialization of the automation environment. The NetView LOADCL command is issued for every program defined by these entries.

If the control file is reloaded, programs that have been previously loaded are not replaced so that the usage counts are kept valid. If a program was previously loaded by this function and is not contained in the current control file, this program is dropped using the NetView DROPCL command.

At initialization (after programs are loaded or dropped), the MAPCL command is issued.

**Usage Note:**

The following example defines the program NEWONE, which AON loads during initialization.

```
RESIDENT NEWONE
```

The SESSION statement is used to identify sessions that are to be actively monitored by AON/SNA.

```
SESSION alias=LU1=lu_name=LU2=lu_name
```

*alias*

The 1–8 character name used to identify the session. Also, use the alias name in an ACTMON statement. Choose names to avoid conflicts with real resources.

*lu_name*

The names used to identify the pair of logical units. The names can be network qualified, as in network_name.lu_name.

**Usage Notes:**

- APPN uses the SESSION control file entry.
- The session must also have an ACTMON control file entry.

**Examples:**
The following example defines a session called TEST1 between LU NT6DI005 and application NTV6D in network USIBMNT.

```
SESSION TEST1,LU1=USIBMNT.NTV6D,
      LU2=NT6DI005
```

The following example defines three sessions. An ACTMON statement for these entries is also required if these sessions are actively monitored.

```
SESSION RJE1,LU1=RALSNDS1,LU2=RALSNDS8
SESSION RJE2,LU1=RALSNDS1,LU2=RALSNDS7
SESSION RJE3,LU1=RALSNDS1,LU2=RALSNDS6
```

---

**SNBU (SNA)**

The SNBU control file entry identifies one or more PUs that are selected for automated speed selection and AON/SNA automated switched network backup (SNBU) using LPDA-2 capable modems. This entry provides the dial phone numbers, the type of notification desired, and other required information for the AON/SNA SNBU automation. The absence of the SNBU entry in the control file indicates that there is no AON/SNA SNBU automation for the PU. The SNBU DEFAULTS parameter is not coded, and AON/SNA SNBU automation does not occur.

You must have a DEFAULTS entry for each PU that is a candidate for AON/SNA SNBU automation. The DEFAULTS entry sets AON/SNA SNBU automation on PUs if AUTOBK=Y. It also tries to use automatic speed selection on all PUs if AUTOSW=Y.

```
SNBU
  RETURNS
    DEFAULTS
      PU
        puname
          LEVEL= 1
          CRIT= N
          MONIT= N
          AUTOSW= Y
          AUTOBK= Y
          ,PH=(num1,num2,numn)
          ,RECONN= N
          ,APO= Y
          POOL= poolname
          FANOUTS=(port1,port2, portn)

DEFAULTS
  Attempts SNBU automation with automatic speed selection on PUs if AUTOSW=Y.

PU
  Defines SNBU backup and speed selection on each PU with the AUTOBK and AUTOSW parameters.

  puname
  Defines the name of the PU to backup.
```
LEVEL
Specifications for AON/SNA to initiate automatic AON/SNA SNBU for Level 1 or Level 2 modems of a tailed circuit configuration. You can define this parameter on a DEFAULTS, PU, or specific PU name.

CRIT
Initiates switched network backup automation when critical thresholds of errors for resources have been reached (PU or LINE). This parameter is mutually exclusive with MONIT1. You can define this parameter on a DEFAULTS, PU, or specific PU name.

MONIT1
Initiates AON/SNA SNBU automation after the first MONIT interval is run. One reactivation recovery cycle has occurred but AON/SNA did not successfully reactivate the resource. This keyword is mutually exclusive with the CRIT parameter. You can define this parameter on a DEFAULTS, PU, or specific PU name.

AUTOSW
Specifies whether you want the modem switched to a lower speed when excessive line alert errors are received. Code AUTOSW as follows:
- Y: Switch modem (recommended default).
- N: Do not switch modem.

You can define this parameter on a DEFAULTS, PU, or specific PU name.

AUTOBK
Specifies whether you want to automate switched network backup:
- Y: Perform backup when a line down alert is received. If you specify yes (Y), you must specify phone numbers with the PH parameter.
- N: Do not perform backup (recommended default).

You can define this parameter on a DEFAULTS, PU, or specific PU name.

PH
Specifies phone numbers of the modem (required if AUTOBK=Y). Enter only digits 0 through 9 for numbers or characters F or P for pauses in the dialing sequence. For example, to specify the dialing sequence 9-1-555-6431 enter 9F15556431. If this is a 2-wire SNBU, enter only one telephone number. Define this parameter for a specific PU only.

RECONN
Enables (Y) or disables (N) automatic reconnection of the leased line after receipt of the second BNJ017I message that is generated by NetView when using IBM 786x model 4y modems.
- Y: Do automatic reconnection on the second BNJ017I message.
- N: Do not automatically reconnect. The operator must use the CHGSNBU command.

Define this parameter for a specific PU only.

APO
Defines the alternate port only. Always select the APO from the pool. This is a required parameter for multipoint lines, but is optional on others.
- Y: Starts AON/SNA SNBU using the first available modem in pool. When you specify yes (Y), you must specify the pool name using the POOL parameter.
N Starts AON/SNA SNBU using the current local modem.

Define this parameter for a specific PU only.

POOL
Defines the name of a pool of modems (NCP lines) available for AON/SNA SNBU for this PU. Must be four or fewer alphanumeric characters.

AON/SNA requires this parameter when APO=Y and requires a corresponding SNBUPOOL parameter. Define this parameter for a specific PU only.

FANOUTS
Defines a list of PUs that are attached to the modem being defined for AON/SNA SNBU automation. Define this keyword for a specific PU only. You can define multiple statements.

Usage:

• If you have a detailed circuit modem configuration, Level 1 modems refer to the local pair of modems, and Level 2 modems refer to the remote pair of modems. AON/SNA attempts automatic switched network backup only on one modem level of a tailed circuit. For example, if you code LEVEL=2, automatic AON/SNA SNBU is only attempted for the Level 2 leased line.

• LEVEL=2 and AUTOSW=Y are mutually exclusive on the same AON/SNA SNBU statement because their combination is not supported by the modems. Therefore, when you code LEVEL=2, also code AUTOSW=N. You can change all AON/SNA SNBU parameters dynamically to run NetView with the SETSNBU command.

• If you specify AUTOBK=Y (explicitly or by default), you must provide phone numbers in the AON/SNA SNBU entry.

• If you specify POOL=poolname, it must match the SNBUPOOL control file entry with at least one member in it. If you specify POOL=poolname and APO=N (explicitly or by default), AON/SNA first attempts to use the same AON/SNA SNBU port that is using the original local modem. Next, AON/SNA attempts to use an alternate AON/SNA SNBU port if the same port is unsuccessful.

• Specify APO=Y on all PUs on multipoint lines, so that only the alternate port AON/SNA SNBU is attempted.

Example:

In this example, system defaults are defined so that each PU is eligible for automatic speed selection due to line quality problems, but none are eligible for automatic AON/SNA dial backup SNBU. Those eligible for AON/SNA dial backup SNBU must have individual entries containing proper phone numbers for the automation to succeed.

SNBU DEFAULTS,AUTOSW=Y,AUTOBK=N,

Example:

In this example, the PU named TA1P26A is eligible for automatic speed selection due to line quality problems and for automatic AON/SNA SNBU. If AON/SNA SNBU is initiated, the call is placed by the existing local modem using extensions 6431 and 6432 to dial the remote modem closest to TA1P26A.

SNBU TA1P26A,AUTOSW=Y,AUTOBK=Y,PH=(6431,6432)

Example:
SNBU (SNA)

In this example, the PU named TA1P26B is eligible for automatic speed selection based on the DEFAULTS parameter. This parameter provides for automatic AON/SNA SNBU. The existing local modem is not used for the AON/SNA SNBU calls. Instead, the first available line and modem in the pool labeled 3174 is selected to call the remote modem closest to TA1P26B, using the numbers shown in parentheses.

SNBU TA1P26B,AUTOBK=Y,PH=(18004445556431,18004445556432),POOL=3174,AP0=Y

SNBUPOOL (SNA)

The SNBUPOOL control file entry is a collection of NCP ports with modems attached, any one of which can be used to back up a PU from a failing link. Each pool defined consists of a SNBUPOOL entry specifying the pool name and one or more poolname-portname pairs that indicate the ports belonging to that pool. Ports may be defined to more than one pool simultaneously.

```
SNBUPOOL

poolname

poolname portname

poolname portname

poolname portname
```

poolname

Specifies a 1–4-character name that identifies a pool of modems. The name must follow the NetView variable naming conventions. This name is also specified on the SNBU entry (POOL=poolname) for each PU eligible to use the pool and on each poolname-portname pair that indicates the members of the pool. This parameter must start in column one of the control file.

portname

Specifies the name of an NCP line (with appropriate modem attached) to be used as an alternate port for SNBU.

Usage:

Alternate port switched network backup requires that there be another line available to move the PU from the failing link. This line must be attached to the same NCP and have similar characteristics as the failed one. In addition, if the alternate line has a PU with the same address as the one being moved to it, the VTAM MOVE command fails. Remember:

• Define an alternate port to the same NCP where the PU is to be moved. This is a VTAM MOVE command restriction.
• If you specify POOL=poolname on a SNBU entry for a PU, you must have a matching the SNBUPOOL control file entry with at least one member in it.
• You may dynamically add or delete the SNBUPOOLS and members for the current run of NetView with the SETPOOL command.

Example:

In the following example, two lines, TA1L100 and TA1L200, are defined as members of the modem pool with the name 3174:

```
SNBUPOOL 3174
3174  TA1L100
3174  TA1L200
```
The following parameters are required to make the AON/SNA SNBU statement valid:

SNBU TA1P26A,AUTOSW=Y,AUTOBK=Y,NPDA=B,PH=(6431,6432),POOL=3174

**SUBSYSTEM (SNA)**

The SUBSYSTEM control file entry identifies the NetView Access Services (NVAS) subsystem that interfaces with the SNA help desk. The SUBSYSTEM control file entry is required only if you are using the NetView Access Services (NVAS) and the SNA help desk for a user ID.

```
SUBSYSTEM subsystemname
```

`subsystemname`

Name of the DB2 subsystem to which NetView connects. `Subsystemname` is a character value with a length of 1 to 4 characters.

**NVAS**

Defines the NetView Access Services (NVAS) subsystem name to the AON/SNA control file.

**JOB**

Defines the VTAM application name of the NetView Access Services (NVAS) procedure with which the SNA Help Desk interfaces.

**DESC**

Provides a short description of the NetView Access Services (NVAS) subsystem. This text is used in some AON/SNA messages.

**Usage:**

Only one NetView Access Services (NVAS) subsystem is supported. AON/SNA must be able to track the reply ID for NetView access services (NVAS) in the status file using the NetView 'MVS' command to interact with the system console.

**Example:**

In the following example, procedure EMS10 is the user-defined procedure to start NetView Access Services:

```
SUBSYSTEM NVAS,JOB=EMS10,DESC='NetView Access Services'
```

**TCP390**

The TCP390 control file entry defines the TCP/IP stack to AON/TCP. A TCP390 entry is required for each stack from which you want to manage.

```
TCP390
```

```
TCP390 MVS_Serv_Pt,IPADDR=xxx.xxx.xxx.xxx,DOMAIN=LOCAL
```
TCP390

MVS-Serv-Pt

Specifies the hostname of the TCP/IP for MVS stack. This name must consist of characters supported by the GLOBALV command. For example, MY_STACK is OK, but MY-STACK is not.

IPADDR=xxx.xxx.xxx.xxx

Specifies the IP address for this MVS service point. The ipaddr is required. You can also specify hostname or both. If ipaddr is not specified, AON attempts to determine the address and adds it to the in-storage copy of the policy definition(s).

DOMAIN=LOCAL | domain

Specifies the NetView domain that owns this MVS service point:

LOCAL Indicates the same host as this AON.

domain The remote NetView domain ID.

FORMAT=STACK

Defines the MVS stack for monitoring.

HOSTNAME=host_name

Specifies the TCP host name associated with this MVS service point. You must specify host_name or ipaddr or both.

HIER2=hier2_appl

Specifies the field in the second alert hierarchy level with a resource type of TP.
HIER3=hier3_device
   Specifies the field in the third alert hierarchy level with a resource type of DEV.

SERVER=(Servername, count)
   SERVER defines the TSO user IDs used to communicate to local service points
   that use the TSO interface. Servername is the root used for the TSO USERIDS.
   Count is the number of TSO Servers. Count will be appended to Servername to
   form a valid TSO SERVERNAME.

UNIXSERV=YES
   UNIXSERV defines this MVS service point as a UNIX environment. This
   parameter must be defined for all service points using UNIX to communicate,
   including local and remote. This is the default.

COMMUNITYNAME
   Defines the SNMP Community name to be used for all AON SNMP requests
   that apply to the stack. The public Community name is the default. The
   community_name can be up to 32 characters in length and is case-sensitive. You
   must use a valid community name for the IP address of your TCP/IP stack.
   Code the COMMUNITYNAME keyword and community_name value on one
   line. Defining the keyword and value on more than one line will cause errors.
   For more information about defining the Community name to TCP/IP, refer to
   OS/390 SecureWay CS IP Configuration.

PINGCNT= 1 | count
   Specifies the number of times to issue the PING. The default is 1.

PINGRETRY= 1 | count
   Specifies the number of PINGS to issue during ACTMON processing.

PINGTIME= 10 | secs
   Specifies the length of time to wait before timing out. The default is 10.

PINGLEN= 64 | length
   PINGLEN is the number of bytes in the echo request. The default is 64.

TCPNAME= tcp/ip proc name
   TCPNAME is the TCP/IP proc name used to specify a TCP/IP address space.
   This parameter must match the TCPJOBname in the TCPDATA file for the
   TCP/IP stack.

SNMPRETRY= 2 | n (SNMP Option)
   Specifies the number of times the SNMP command will be retried. Valid values
   are in the range of 1–10. The default is 2.

SNMPTO= 3 | n (SNMP Option)
   Defines the number of seconds that SNMP will wait for a response. Valid
   values are in the range of 1–30. The default is 3.

MAXREP= 10 | n (SNMP Option)
   Specifies the number of lexicograms to be returned for each variable binding
   pair. Valid values are in the range of 1–99. The default is 10. Valid only when
   Bulk is a selected option.

NONREP= 0 | n (SNMP Option)
   The number of binding pairs for which only one successor lexigram is
   returned. Valid values are in the range of 0–99. The default is zero (0). Valid
   only when Bulk is a selected option.
TCP390

RPLENGTH= 64 | n (SNMP Option)
The length of the ICMP message to be sent to the remote host. Valid values are
in the range of 16–4096. The default is 64. Valid only for the remote ping
function.

RPTO= 5 | n (SNMP Option)
The time that the Remote Ping function will wait for a response in seconds.
Valid values are in the range of 3–15. The default is 5. Valid only for the
Remote Ping function.

DVIPA=Y|N
Provides the user with the ability to limit which stacks are DVIPA/Sysplex
Distributor capable and which stacks should be managed by AON/TCP for
DVIPA.

Y Y(es) indicates that AON/TCP gathers MIB data from each of the
stacks during the DVIPA polling cycles and provides DVIPA/Sysplex
Distributor management functions. This is the default.

N N(o) indicates that the stacks will not participate in DVIPA/Sysplex
Distributor functions or will not be managed by AON/TCP.

You can use No to minimize CPU cycles.

MAXCONN= nnn
Specifies the maximum number of connections to display at the Web browser.

MAXDVIPA= nnn
Specifies the maximum number of DVIPAs to display on the DVIPA Definition
and Status panel at the Web browser.

Usage Note:
You must specify host_name or ipaddr or both.

TCPIP
The TCPIP entry defines critical Tivoli NetView for UNIX network resources to
AON/TCP. If you specify FILTER=Y on the NV6000 statement, AON/TCP
manages only the nodes defined with the TCPIP statement. Any nodes that you
want to actively monitor with the ACTMON definition require a TCPIP definition.

TCPIP

alias Indicates a customer chosen name for a TCP/IP resource. This name can
indicate the function of the device, location, owner, or any other
meaningful unique identification. The first word of the hostname (all
characters to the first dot) is often a good choice for an alias. The alias
used must also be a valid REXX variable name.
SP Identifies which Tivoli NetView for UNIX Service Point to use when issuing TCP/IP commands. A NV6000 control file statement is required for this service point.

RESTYPE Indicates one of the following valid resource types:
- HOST
- INFC
- NAMESERV
- IPRouter
- LINK

IPADDR Consists of dotted decimal IP addresses up to 15 characters including dots. They must be in the 256.256.256.256 format. You cannot use any alphabetic characters.

Because some hosts have multiple IP addresses, you can define a list of IP addresses. If a failure for any of these IP addresses is received, failure processing is done for the host. Because AON/TCP uses the first IP address in the list as a target for doing pings, put the most important or frequently used IP address first in the list.

HOSTNAME Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. Valid characters for Hostname are 1 through 9, A through Z, #, @, and &. Hostname is case sensitive. The following is an example of a hostname:

```
mrpres.whitehouse.capital.usa
```

LINK Indicates one or multiple links that act as interfaces to other TCP/IP network gateways, routers, or hosts. This keyword is valid only for a resource type of router. For this list, you can use the IP address, the last word of the host name, or the alias. Use all uppercase letters in this definition.

Note: Defining resources with TCPIP definitions ensures that the operators see information about that resource with a consistent name.

TGSwitch (SNA)

The TGSwitch control file entry switches traffic load back to the primary line where the load is recovered. VTAM automatically switches to the backup line in the event of a failure on the primary line. Because VTAM does not switch back to the primary line after it is recovered, TGSwitch is used to return to the primary line by inactivating and reactivating the backup line when the primary line becomes active.

This is especially important when the two lines are of unequal speeds. Then, the high-speed line should be the primary data carrier.

TGSwitch

```
——TGSwitch—— prim_line——,BKUP=bkup_line———
```

prim_line
The primary line, usually the higher speed.
TG SWITCH (SNA)

**BKUP**

The backup line, usually the lower speed

**Usage:**

Both lines should be part of the same VTAM transmission group (TG).

**Example:**

This example assumes that lines TA1L1004 and TA1L1005 are part of a VTAM transmission group (TG) and that the bulk of the traffic over this TG should go by way of TA1L1004.

TGSWITCH TA1L1004,BKUP=TA1L1005

**THRESHOLDS**

The THRESHOLDS control file entries specify the number of times an event must occur to define infrequent, frequent, and critical error situations. A THRESHOLDS DEFAULTS entry is required in the control file.

If the number of errors exceeds the number defined in the THRESHOLDS entry, AON notifies the operator. This allows operators the opportunity to check on resources that are experiencing degrading availability before a failure occurs. If a resource exceeds a critical threshold, recovery processing typically is stopped.

Multiple THRESHOLDS entries may be included in the control file, one for each resource, one for the resource type, and another for the default actions. Any combination of THRESHOLDS entries can be included. The most specific THRESHOLDS entry found determines which thresholds are used.

AON is designed to meet most needs of thresholds automation processing. However, your installation may have special requirements that require you to tailor threshold automation processing. You can create AON user exits for threshold values for a specific resource. These values are specified in the THRESHOLDS entry or in the ENVIRON EXIT entry.

```
THRESHOLDS

res_name
    CRIT=(nn,hh:mm,Y|N), FREQ=(nn,hh:mm)
    INFREQ=(nn,hh:mm), EXIT06=(User_Exit)
    FAILURE=(tcp_retype_list, ALL)-, AUTH=(cnt,intvl, ALL)
    DISK=disk_percentage, CPU=cpu_percentage
```

*res_name*  
The name of the resource to be automated. This resource name can be a network resource type, resource name, or resource name mask (for an
installation’s naming conventions). Examples of resources names are PU01 and LINE23. This is the most specific type of THRESHOLDS entry.

**res_type**
The type of the resource to be automated. The resource type defined for AON is for network components. Examples of resource types are LU, PU, LINE, CDRM, NCP, APPL, and LINKSTA. The following wildcard characters can be used:

- `*` Multiple character wildcard
- `%` Single character wildcard

For example, both PU0* and P%%1 match for PU01.

**res_name**
The group of resources levels to be automated

**DEFAULTS**
Used when neither the resource name or resource-type are specified. This is the least specific THRESHOLDS entry. A THRESHOLDS DEFAULTS entry is required in the control file.

**CRIT**
Specifies the critical threshold occurrences and time interval. Typically, recovery attempts are stopped if a resource reaches its critical threshold. This helps to prevent potentially recursive situations and to stop recovering resources that do not stay active long enough to provide usability for end users.

- `nn` Number of occurrences in the range of 1–10 within the time period specified in
- `hh:mm` Time period in hours and minutes (hh:mm) in the range of 00:01–99:59.
- `Y|N` Specifies if the critical threshold exception should invoke the CRITTHRS Notification Policy. The default is Y or Yes.

**FREQ**
Specifies the frequent threshold occurrences and time interval.

- `num` Number of occurrences (from 1 to 10) within the time period specified in `intv` that define the frequent threshold.
- `intv` Time period in hours and minutes (hh:mm) for the frequent threshold occurrences specified in `num` in the range of 00:01–99:59.

**INFR**
Specifies the infrequent threshold occurrences and time interval.

- `num` Number of occurrences in the range of 1–10 within the time period specified in `intv` that define the infrequent threshold.
- `intv` Time period in hours and minutes (hh:mm) for the infrequent threshold occurrences specified in `num` in the range of 00:01 – 99:59.

**EXIT06**
Specifies the user exit information for AON to use instead of what is specified in the ENVIRON EXIT control file entry.

**FAILURE**
Indicates the following:

- **TCP_restype_list**
  Indicates to which resource types thresholding analysis takes place. More
THRESHOLDS

than one resource type can be listed. Because alerts flow from a TCP/IP resource in groups when connectivity is lost or established, one or more resource types can be chosen as the significant type for the TCP/IP resource. For example, if a TCP/IP resource loses connection to the network, an alert flows for each interface card (INFC), the host (HOST), and perhaps the link (LINK), as well.

If FAILURE=(HOST,THR), only failures for the host (including the host name) are analyzed for threshold exceptions, and operators are notified only if the failure causes a threshold exception. If you specify several resource types, you must increase the threshold counts to accommodate the fact that each failure increases the threshold failure count by more than 1. If FAILURE=(HOST,ALL), operators are notified each time a host fails.

THR|ALL
Defines whether an operator is to be notified every time a resource of this resource type has a failure (ALL), or only when a threshold for the resource of this resource type has been exceeded (THR).

AUTH
Indicates the following:

cnt \(1-99\)
Indicates how many security authorization failures are allowed to occur for a resource within the interval allotted time period before the security authorization failure threshold is exceeded. The limit is 99. When the security authorization failure threshold is met, operators are notified of the condition.

intvl
Defines the interval of time, in the hh:mm format, in which AON/TCP is counting security authorization failures for thresholding. The variable hh can be in the range of 0–99, and mm can be in the range of 0–59.

THR|ALL
Defines whether the operator is to be notified every time there is a security authorization failure (ALL) or only when a security authorization failure threshold for the resource has been exceeded (THR).

DISK
Indicates whether an alert has been received from a TCP/IP host reporting a disk utilization percentage that exceeds the amount defined in disk_percentage. AON/TCP notifies operators of a disk utilization threshold exception. This value can be 0–99.

CPU
Indicates whether an alert has been received from a TCP/IP host reporting a CPU utilization percentage that exceeds the amount defined in cpu_percentage. AON/TCP notifies operators of a CPU utilization threshold exception. This value can be in the range of 0–99.

Usage:

The THRESHOLDS entries should uniquely describe consistent intervals for processing, with the critical thresholds having either the smallest interval or the highest frequency of occurrence, followed by frequent, then by infrequent conditions.
AON uses only one THRESHOLDS entry to determine if the thresholds have been exceeded. The first THRESHOLDS entry that AON finds is the one it uses. The THRESHOLDS entries are searched in the following order:

1. Resource name
2. Resource type
3. DEFAULTS

The easiest method is to use the THRESHOLDS DEFAULTS entry as much as possible. Use the THRESHOLDS resource or resource-type only when they must be different than the THRESHOLDS DEFAULTS entry.

Example:

In this example, system defaults are defined as follows:

**Critical threshold**
Three errors within one hour

**Frequent threshold**
Three errors within two hours

**Infrequent threshold**
Two errors in eight hours

THRESHOLDS DEFAULTS,CRIT=(3,01:00),FREQ=(3,02:00),INFR=(2,08:00)

Example:

In the following example, PU thresholds are defined as:

**Critical threshold**
Two occurrences within 45 minutes

**Frequent threshold**
Two occurrences within four hours

**Infrequent threshold**
Two occurrences within 24 hours

THRESHOLDS PU,CRIT=(2,00:45),FREQ=(2,04:00),INFR=(2,24:00)

**TIMER**

The TIMER control file entries define default timers that initiate commands or command lists at scheduled times. The timer can be scheduled for a specific day and time, after a certain day of the year, or repetitively at intervals. Multiple TIMER entries can be specified in the control file, one for each scheduled command.

The TIMER entry is optional. If a TIMER entry is not included in the control file, the online TIMER facility can be used to schedule timer events. The TIMER command provides a full-screen operator interface for displaying, adding, deleting, and modifying timers for NetView.
**TIMER**

```
+,SAVE=NO+,CATCHUP=NO+,COMMAND='command'
```

*timer_id*

The identifier that you specify for this timer request. The *timer_id* can be from one to eight characters in length. You cannot use the following for *timer_id*:

- The word ALL
- The characters EZL, FKV, FKW, FKS, SYS, or RST as the first three characters
- Any VTAM resource name or Dnccname or Rnccname.

**Note:** If you use any of the above, automation may be unpredictable.

**TIME**

Specifies when or how often you want the command issued.

**AT**

AON issues the command at the specified time. The following rules apply when validating the date or time of the AT parameter:

- If the date defaults and the time specified is earlier than the current time, AON schedules the command for the next day.
- If \( mm/dd \) is the same as the current month and day, the time must be later than the current time or AON issues a notification message.

**EVERY**

AON issues the command repetitively at a timed interval. The EVERY parameter is not valid with a date.

**AFTER**

AON issues the command after a specified period of time.

**#days**

Specify the number of days \( ddd \) with the EVERY or AFTER parameter as \( ddd \) in the range of 1–365. If you specify \( ddd \), you must also specify a start time.

**weekday**

Specify weekday with the AT or EVERY parameter as a day of the week, or DAILY.

**date**

Specify a date with the AT parameter as \( mm/dd/yy \), where \( mm \) is the month, \( dd \) is the day of the month, and \( yy \) is the year. The default value is the current date. AT is not valid with DAILY.

**start**

Specify the start time as \( hh:mm:ss \), \( hh:mm \), or \( mm \) where \( hh \) is the hours in the range of 00–23, \( mm \) is the minutes in the range of 0–59, and \( ss \) is the seconds in the range of 0–59. The default is 00.

**TASK**

Specify where the command is to run. You can specify:

*oper_id*

The NetView operator identification under which this timer runs. If you specify *oper_id*, AON issues the timer command only if the operator is logged on.

**PPT**

Specify that the command or command list indicated by the "COMMAND"
parameter should run under the PPT. If you specify PPT, the command runs regardless of which operators are logged on at the indicated time.

---

**Notes about the PPT**

Not all commands can run under the PPT. Examples of commands that cannot run under the PPT are:

- Commands that control the screen (such as AUTOWRAP, INPUT, and SET PF<sub>nn</sub>).
- Commands or command lists that start full-screen command processors (such as BGNSESS FLSCN, NLDM, NPDA, or BROWSE).
- Commands that issue STIMER.
- Command lists that issue the control statement &WAIT or &PAUSE
- REXX command lists that issue WAIT, PAUSE, or TRAP.

---

**autoop**

The operator ID as specified on the AUTOOPS control file entry. For more information, see "AUTOOPS" on page 214.

**SAVE**

Indicates to NetView whether this timer event should be saved to the NetView SAVE/RESTORE database. If SAVE is not specified, the timer event is not saved. SAVE=YES is required if CATCHUP=YES is specified.

**CATCHUP**

Allows a timer that has been saved to be caught up after a system outage occurs. If the system outage occurs prior to a timer being started and restarts after the timer should have started, specifying CATCHUP=YES allows the timer command to be issued after the specified time. This parameter is valid only with the AT operand.

**COMMAND**

The command or command list AON runs when the timer expires.

**Usage Notes:**

If SAVE=YES is specified for a timer and the timer data is changed at a later date (leaving the timer-id the same) in the control file, the operator must also change the data for the timer-id through the operator interface. The timer-id is restored from the SAVE/RESTORE data base rather than reset from the control file. The last saved data may be old data if it is not updated online before NetView shutdown.

Timers are not reset when the control file is loaded. If timer definition data in the control file is changed, you must use the TIMER command to alter the current TIMER settings in NetView.

Commands defined as REGULAR or BOTH when NetView was installed may be used with TIMER. Commands defined as IMMEDIATE cannot be used with TIMER.

If you specify the EVERY parameter with CATCHUP=YES, the timer is not scheduled. If you specify the EVERY parameter, the command runs at the indicated interval until you purge the EVERY command.
TIMER

To avoid overloading system resources, you should not schedule an excessive number of commands at short time intervals.

If you schedule commands under the PPT, they may not run in the order that you specify if the value of the time parameter is the same for each command.

Example:

In this example, AON initiates the CDRMS command list on September 28, 1999 at 5:20 PM The timer initiates only if operator NETOP2 is logged on. If a system failure occurs prior to 17:20 and restarts after 17:20, this timer initiates because CATCHUP=YES is specified.

```
TIMER RUNCLST,TIME='AT 09/28/99 17:20',SAVE=YES,TASK=NETOP2,
                   CATCHUP=YES,COMMAND='CDRMS'
```

Example:

In this example, AON issues the command at 5:21 PM on Thursday to load NCP21. If there is a system outage, this timer does not CATCHUP because SAVE=NO is specified. Operator NETOP2 must be logged on for this timer to start.

```
TIMER LOADNCP,TIME='AT THURSDAY 17:21',SAVE=NO,TASK=NETOP2,
                  CATCHUP=YES,COMMAND='V NET,ACT,ID=NCP21,LOAD=YES'
```

Example:

In this example, AON issues the D NET,BFRUSE command at noon on every Sunday. The operator BJSIMPS must be logged on. This timer does not CATCHUP because the EVERY parameter was specified.

```
TIMER DISPLAY1,TIME='EVERY SUNDAY 12:00',TASK=BJSIMPS,
             CATCHUP=YES,COMMAND='D NET,BFRUSE'
```

Example:

In this example, AON issues the D NET,CDRMS command after 20 hours and 40 minutes. Because there is no task keyword defined, this timer runs if the operator who loaded the control file is logged on.

```
TIMER DISPLAY2,TIME='AFTER 20:40',SAVE=YES,
               COMMAND='D NET,CDRMS'
```

Example:

In this example, AON issues the MVS D J,L command every 150 days at 12 noon. This timer is not to be saved in the SAVE/RESTORE data base.

```
TIMER MVSDISP,TIME='EVERY 150 12:00',SAVE=NO,TASK=NETOP1,
             COMMAND='MVS D J,L'
```

Example:

In this example, AON issues the command at 6:15 PM on October 2, 1999, to print a dump. This timer initiates under the PPT and catches up if there was a system outage during its scheduled time.

```
TIMER DISPLAY3,TIME='AT 10/02/99 18:15',SAVE=YES,TASK=PPT,
             CATCHUP=YES,COMMAND='D NET,ID=CICSAPPL'
```

Example:
In this example, after 6 minutes, AON issues the `VARY NET,ACT,ID=CDRMAB` command.

```
TIMER DISPLAY4,TIME='AFTER 6',SAVE=YES,
    COMMAND='VARY NET,ACT,ID=CDRMAB'
```

Example:

In this example, the `EZLEOIVT` command automatically removes resources from the OIV view when the resources’ display status is updated to satisfactory (129).

```
TIMER ADOIV,TIME='EVERY 00:03',
    TASK=AUTOIV1,
    COMMAND='EZLEOIVT'
```

Example:

In this example, AON issues the `MVS S SYSLOG` command every 20 minutes. This timer runs on the operator ID performing Timer initialization. Normally, AUTO1 performs initialization during AON startup, therefore, this timer is scheduled to initiate under AUTO1 if this entry is defined in the control file.

```
TIMER MVSSRT,TIME='EVERY 20',SAVE=YES,
    COMMAND='MVS S SYSLOG'
```

---

**TSOSERV**

Defines the name of the MVS procedure to use when starting the TSO servers.

```
TSOSERV server PROC=.procname
```

**server**

The root name of the TSO server from the TCP390 SERVER parameter.

**procname**

The name of the catalogued procedure to be used for this TSO server. The default is CNMSJTSO.

Usage Notes:

- Different MVS procedures are used to point the TSO server to the TCPDATA for different TCP/IP stacks:
  - One stack can use:
    ```
    //SYSTCPD DD DSN=tcp32.init(TCPDATA)
    ```
  - A second stack can use:
    ```
    //SYSTCPD DD DSN=TCP32.INITB(TCPDATA)
    ```
- The server parameter is only used for locally attached TCP/IP stacks. To define TSO servers for remote NetView programs that are not running AON, refer to [Tivoli NetView for z/OS Installation: Configuring Additional Components](#).

---

**X25MONIT (SNA)**

The X25MONIT control file entry defines which AON/SNA X.25 switched virtual circuits (SVCs) to monitor. You can display AON/SNA X.25 switched virtual circuits (SVC) by using the X25MONIT command.
X25MONIT LINE

\[\text{X25MONIT LINE}: n, \text{X25MCH}=\text{linename}, \text{MCHGRP}=\text{groupname}, \text{LOLCN}=\text{lowest}_\text{ln}c, \text{HILCN}=\text{highest}_\text{ln}c, \text{CALLTYPE}=\text{IN/OUT}, \text{TIMER}=\text{min}, \text{THRESH}=\text{count}\]

\[n \ \text{Specifies a sequential line count, 1 to } n.\]

X25MCH

\[\text{Specifies the } \text{linename} \text{ on the X25MCH macro.}\]

MCHGRP

\[\text{Defines the NCP } \text{groupname}, \text{ which contains the X.25 definitions.}\]

LOLCN

\[\text{Defines the } \text{lowest}_\text{ln}c, \text{ which is the lowest logical channel number (LCN) in the group.}\]

HILCN

\[\text{Defines the } \text{highest}_\text{ln}c, \text{ which is the highest LCN (logical channel number) in the group.}\]

CALLTYPE

\[\text{Defines the characteristics of a logical group channel. The parameter choices are IN, OUT, and INOUT.}\]

TIMER

\[\text{Sets a timer to monitor switch virtual circuits (SVCs) status at the specified number of minutes (1–99 minutes).}\]

THRESH

\[\text{Sets the minimum number of free switched virtual circuit (SVCs) for which the user wants to be notified. The number must be less than the number set in the HILCN and LOLCN parameters.}\]

Usage:

- Specify the LOLCN and HILCN parameters in hexadecimal and include the logical group channel number when it is different from 0. Define the LOLCN and HILCN parameters as \(x^g\text{nn} \text{ and } \text{nn}=\text{lc}n\).
  
  \[\text{where}\]
  
  \[g \ \text{The logical group channel number.}\]
  \n  \[\text{nn} \ \text{The lowest logical channel number operand.}\]
  
- Define the X25MCH and MCHGRP parameters with the following NPSI naming convention:
  
  - X25MCH must be \(pL\text{xxllx} \text{c}n\)
  
  - MCHGRP must be \(p25S\text{xxxy}\)
  
  \[\text{where}\]
  
  \[p \ \text{The prefix coded on the X25BUILD macro.}\]
  \n  \[\text{xxx} \ \text{The address on the ADDRESS operand of the X25MCH macro.}\]
  \n  \[\text{lc}n \ \text{The logical channel number operand of the X25MCH macro.}\]
y The group sequence on the physical length.

- AON/SNA performs passive monitoring for all X25MCHs that you define. AON/SNA performs active (TIMER) monitoring for X25MCHs with TIMER=min.
- Use the X25MONIT command to see the results of the monitoring. You can use the X25MONIT command to update these definitions dynamically.

**Example:**

For line one, HILCN is 08 and LOLCN is 02. This equates to seven total switched virtual circuit (SVCs) with no sessions started, so the value of free switched virtual circuit (SVCs) should be seven. As sessions start, the value of free switched virtual circuit (SVCs) is decremented, and when the value falls to 03, a message is issued to notify the system that the threshold has tripped.

Timer=10 indicates that every ten minutes, AON/SNA checks the status of the components of this line.

```
X25MONIT LINE1,
   X25MCH=XL01001,
   MCHGRP=X25S01B,
   CALLTYPE=INOUT,
   LOLCN=02,
   HILCN=08,
   TIMER=10,
   THRESH=3
```
Chapter 4. Inform Policy Member

The INFORM policy member is used to define which personnel should be contacted, when they should be contacted and how they should be contacted. By default the INFORM policy provides support for E-mail, numeric and alphanumeric pagers. An inform sample member (EZLENETF) is provided which enables these actions through a NetFinity service point. However, the sample is designed to be modified, and exits are provided so existing technologies can be used. Using the SETUP, GROUP, INFORM and CONTACT statements that follow will define your INFORM policy member. See the INFORMTB command in the Tivoli NetView for z/OS Command Reference for more information. Refer to the Tivoli NetView for z/OS Automated Operations Network User's Guide for more information on the Inform Log Utility (ILOG).

Use the SETUP, GROUP, INFORM and CONTACT statements that follow to define your INFORM policy member.

Usage Notes:
- An asterisk in column 1 denotes a comment line in the member.
- Columns 1–72 contain the statements. Columns 73–80 are truncated.
- A semicolon is used to denote the end of a statement.
- A comma is used to separate the keyword and value pairs.
- Continuation is assumed until a semicolon is detected.
- The INFORM policy member supports the use of %INCLUDE statements. However, it is recommended that you only use these statements before, between, and following completed inform member statements. Using %INCLUDE statements from within inform member statements will produce unpredictable syntax error messages, should the %INCLUDE statement fail.
- For interfaces other than EZLENETF it may be necessary to place dummy values in the inform policy member for some keywords, such as, SP=.
- SMTP can be used for E-mail only.
- The inform logging function is not intended to be used as a general purpose log. Messages resulting from inform actions are logged in the NetView log for that purpose. ILOG is intended for use by those who want to actively acknowledge, track, and delete inform records. Due to the I/O required, Inform logging is not recommended in all installations.

SETUP

The SETUP entry in the INFORM policy member is used to set several key inform definitions. These definitions are used both when the inform member is loaded and during runtime. The settings allow definition of additional connection types, enablement of the inform log support, specification of the service point or application domain, and the specification of the inform exit routines. The SETUP entry must precede the first INFORM entry, and there can only be one SETUP entry. Also, the SETUP statement enables the use of the INFORM policy member without initializing AON.
SETUP

CONNECTIONS
A list of additionally supported connection types. If an interface supports FAX, then FAX must be added to the SETUP CONNECTIONS parameter. It is not necessary to add NUMPAGE, ALPHAPAGE, or EMAIL to this list.

LOG
Set the LOG keyword to YES if you want the INFORM log enabled. When enabled, selected INFORM actions are logged, and this list can be displayed using a full screen AON function named ILOG. ILOG allows each INFORM action to be deleted, acknowledged, or reissued as needed. Because I/O is required the default for SETUP LOG is NO. Actions caused by the use of the INFORM (EZLECALL) command are not logged. All actions generated by enabling the control file NOTIFY entry with the INFORM keyword enabled, are logged. All calls to EZLENRM are logged. Calls made by operators using EZLECALL are not logged. See LOGCALLS to enable these.

MEMBER
If LOG=YES is specified, then membername is the member that AON INFORM uses to write the log records. The records are written to the first data set name found in the DSILIST data set definition. If no member name is specified, EZLIFLOG is used by default. The write protection key >INFORM on the first line of the member starting in column 1 is written. Therefore, the INFORM log function overwrites only other INFORM logs.

LOGTASK
The INFORM log requires an autotask for the sequencing of all updates. If the log is enabled, the autotask indicated is started, or AUTOINF is started by default.

LOGCALLS
Set LOGCALLS to YES if you want to log both automation actions and operator inform actions. It is recommended that only NOTIFY actions be logged. If needed, operator INFORM actions can also be logged by setting LOGCALLS=Yes.

SPDOM
The domain ID of the network NetView that owns the INFORM service point or application. The current domain is the default.

EXIT11
The preprocessing exit for the INFORM policy. Exit 11 is invoked just prior to
consulting the in-storage INFORM policy information. Exit 11 can be used to
dynamically change the INFORM policy name, group name, or provide a list
of policy and group names.

The inputs from EZLENFRM are:

- policy name
- resource name
- resource type
- domain id
- resource status
- automation status id

The inputs from EZLECALL are:

- policy name
- domain id

Exit 11 can affect changes in the flow, based on the following return codes:

0  Continues normal INFORM policy processing.
4  Continues normal INFORM policy processing using a new INFORM
    policy name, group name, or a list of policy names. The new
    information is passed back to the INFORM flow using a task global
    variable with the following naming convention: EZLPOLEX.
    EZLPOLEX=policy name|group name|list of comma-delimited policy
    and group names.
8  Discontinue processing. Allows linkage to other INFORM technologies
    without returning to this INFORM policy flow.

EXIT12

A postprocessing exit for the INFORM policy. Just prior to invoking the
interface specified in the policy, and passing the entire policy which includes
the error or operator message, Exit 12 is invoked. The inform action (as
specified) can be cancelled. If changes are required, the inform action must be
cancelled and the updated call to the interface made by the exit routine.

The parameters passed to the INTERFACE are passed to Exit 12. Refer to the
EZLENETF sample for instructions on the format of these parameters.

This exit can affect changes in the flow based on the following return code:

0  Call the interface code as specified in the INFORM policy CONTACT
    statement.

n  For values other than zero (0), processing is discontinued. For instance,
you might want to alter phone numbers or service points at this point
in the flow. To do this, alter the necessary fields and call the
appropriate interface routine directly.

Note: Exits can be written in any language supported by Tivoli
NetView for z/OS. Where applicable, compile exits and load
them into storage.

Example:

This example will add a setup statement, to define other connection types and
enable logging. It will specify the log member name as INFLOG, by default
EZLIFLOG would be used. Although a customer written interface would be
required to implement the FAX connection:

```
SETUP CONNECTIONS=FAX,LOG=YES,MEMBER=INFLOG;
```
The GROUP entry in the INFORM policy member is used to specify a list of policies that form a logical contact group. For example, if you want to define an INFORM policy for each person who may be contacted. Using these definitions, each person can then be added to any number of different groups. This eliminates the need to duplicate CONTACT statement definitions for individuals. GROUP statements may be coded anywhere in the INFORM policy member except between an INFORM statement and its corresponding CONTACT statements.

**GROUP**

- **GROUP group_name LIST=policy_name**

  *group_name*
  
The name of a logical grouping of other groups or INFORM policy names.

  *policy_name*
  
The policy name used to identify the list of the following contacts. The *policy_name* is referenced by the NOTIFY statements in the control file, or by the INFORM command.

**Note:** A GROUP statement which contains other group names may do so only if these groups were previously defined. Also, a group cannot refer to itself.

**Example:**

This example will define groups so contact statements do not need to be duplicated in multiple policies. Note: PersonC and CompanyA are listed in multiple groups:

GROUP DAYOPS, LIST=PERSONA, PERSONB, COMPANYA;
GROUP NITEOPS, LIST=COMPANYA, PERSONC;
GROUP WEEKENDS, LIST=PERSONC, PERSOND;

**INFORM**

The INFORM entry in the INFORM policy member is used to denote the starting point of a new policy. An INFORM policy contains a list of contacts and how and when those contacts should be notified. An INFORM statement must be followed by one or more CONTACT statements.

**INFORM**

- **INFORM policy_name,** SP=spname, SPDOM=spdomain**

  *policy_name*,
  
  The policy name used to identify the list of the following contacts. The *policy_name* is referenced by the NOTIFY statements in the control file, or by the INFORM command.

  *spname*, *spdomain*
INFORM

policy_name
The policy name used to identify the list of contacts which follow. policy_name is referenced by NOTIFY statements in the control file, or by INFORM commands.

SP The SNA service point or application name to be used for this policy, unless overridden by the individual contact statements. This is an optional parameter. Up to 80 characters can be used to define this field.

SPDOM The network NetView domain where the INFORM action service point or application is owned. If this optional parameter is not specified, the current domain is assumed. If assigned on the SETUP statement, the value specified here is used.

COMPORT The communications serial port to be used on the service point unless overridden by the individual contact statements. Comport is a required parameter when CONNECTION is ALPHAPAGE or NUMPAGE. However, if it is not specified here it must be specified on the CONTACT statement.

Example:

This example defines an Inform policy called COMPANYA. The service point is specified as NT011002. The service point domain defaults to the current domain and the comport is specified on the required contact statements that follow (not shown here).

INFORM COMPANYA, SP=NT011001;

CONTACT

The CONTACT entry in the INFORM policy member defines the criteria used by AON to determine who is contacted for a given policy. An INFORM statement must precede a contact or list of contact statements.


**Notes:**

1. TAPACCESS is only required when CONNECTION=ALPHAPAGE is specified.

**ONCALLDAY**

Specifies the day or days this contact is on call.

* Specifies the contact is on call seven days a week. The asterisk (*) is the default.

**WEEKDAY**

Specifies the contact is on call Monday through Friday.

**WEEKEND**

Specifies the contact is on call Saturday and Sunday.

* dayofweek

  Specifies the day of the week.

  MONDAY|TUESDAY....|SUNDAY

**ONCALLTIME**

Specifies the time the contact is on call during the selected ONCALLDAY.

* Specifies the contact is on call 24 hours a day. The asterisk (*) is the default.

* hh:mm

  Specifies the on call start time, where hh is the hour in the range of 00–24, and mm is the minutes in the range of 00–59.

  to  Separator required between the start time and the stop time.
Specifies the on call stop time, where \textit{hh} is the hour in the range of 00–24, and \textit{mm} is the minutes in the range of 00–59. The stop time must be later than the start time.

\textbf{Note:} 24:00 hours represents midnight and is supported for readability of the policy.

\textbf{CONNECTION}

Specifies the method to be used when contacting the specified personnel. Additional CONNECTION types that support customer–written interfaces must be added to the SETUP statement connection keyword in this member. CONNECTION type names must be 1–10 characters.

\begin{itemize}
  \item \textbf{ALPHAPAGE} Alphanumeric pager
  \item \textbf{NUMPAGE} Numeric pager
  \item \textbf{EMAIL} Electronic mail
\end{itemize}

\textbf{ROUTE}

Specifies the phone number to be used to contact the ALPHAPAGE or the NUMPAGE. If the connection type is EMAIL, then the ROUTE parameter contains the e-mail address. A maximum of 80 characters is supported.

\begin{itemize}
  \item \textit{phone\_#}
    \begin{itemize}
      \item A string of integers containing all access numbers, area code, and phone number.
    \end{itemize}
  \item \textit{email\_address}
    \begin{itemize}
      \item A continuous string of characters that includes the user ID and address. For example, SantaClaus@NORTH.POLE.COM.
    \end{itemize}
\end{itemize}

\textbf{MSG}

Specifies the message that will be sent. The message type and format is dependent upon the CONNECTION type selected.

For \texttt{CONNECTION=ALPHAPAGE} or \texttt{CONNECTION=EMAIL}, the message is a string of up to 80 characters. AON provides a default message, or a custom message can be provided with synonym replacement as follows for calls made through \texttt{EZLENFRM}:

\begin{itemize}
  \item \texttt{%mm\%} Month
  \item \texttt{%dd\%} Day
  \item \texttt{%yy\%} 2–digit year
  \item \texttt{%yyyy\%} 4–digit year
  \item \texttt{%hr\%} Hour
  \item \texttt{%mn\%} Minute
  \item \texttt{%policynamE\%} Replaced with the name of the current policy
  \item \texttt{%resname\%} Replaced with the resname
  \item \texttt{%restype\%} Replaced with the restype
  \item \texttt{%resdomain\%} Domain of the failing resource
  \item \texttt{%resstat\%} Replaced with the resources status at time of failure
  \item \texttt{%aostat\%} Replaced with the automation status
\end{itemize}
The + character is used to continue a message line in the policy member to the next line. When the message line is combined, a blank is substituted for the +. If the + is omitted, the message line will be combined with no space between the data on the two lines.

default

The following is a default message:

INFORM FOR %restype% %resname% STATUS=%resstat% DOMAIN=%resdomain%

Note: Message length must be handled by the INTERFACE. For NetFinity, 80 characters will be supported for alerts sent as e-mail, and 60 characters for alphanumeric pagers.

For CONNECTION=NUMPAGE, the message must be a phone number.

phone#

String of integers including all access numbers, area code, and phone number.

Note: Phone numbers can be specified as follows: 1–(919) 123–4567. The following characters will be removed: blanks, parentheses, commas and dashes. After these characters have been removed, the resulting phone number can be up to 20 numeric digits.

Note: Coded call-back numbers, use the numeric message to store a numeric code versus a call-back number. Use EXIT12 to alter the coded number.

NAME

The NAME field can contain up to 40 characters such as the name, group, or company to be contacted.

INTERFACE

The INTERFACE field specifies the name of the routine that implements the connection to the specified contact platform.

EZLENETF, EZLESMTP or another valid command name

COMPORT

The COMPORT field can contain up to 8 characters representing the communications port name where the modem is attached to the service point. If no name is specified, the default COM1 is used. The COMPORT value specified on the CONTACT statement overrides the COMPORT value specified on the INFORM statement.

TAPACCESS

The Telocator Alphanumeric Protocol access number is required when contacting an alphanumeric pager. As many as 20 integers can be used to specify the access code and phone number.

Note: Phone numbers can be specified as follows: 1–(919) 123–4567. The following characters will be removed: blanks, parentheses, commas and dashes. After these characters have been removed, the resulting phone number can be up to 20 numeric digits.

SP

The SP field specifies the SNA service point or application name for the INFORM policy entry. If no SP name is specified, the SP name specified on the INFORM statement will be used. An SP name is required on each CONTACT
statement or on the INFORM statement. The SP field can contain up to 80 characters. For customer interfaces, it may be necessary to set the SP keyword to a dummy value.

**SPDOM**
The SPDOM field specifies the domain ID of the network NetView that controls the service point or application.

**PARMS**
The PARMS field provides a space so additional parameters can be sent to customer–written interfaces. Up to 100 characters will be passed to the interface routine that provides the error checking and parsing. For example, additional message text can be passed in this field.

**Example:**

This example will use the inform policy for PersonC. PersonC is available every evening by e-mail, and additionally by Pager on the weekends. The service point specified in the inform statement applies to both contact statements. The interface defaults to EZLENETF for both contact statements:

```
INFORM PERSONC,SP=SP000002;
CONTACT ONCALLDAY=*,
  ONCALLTIME=16:00 to 24:00,
  CONNECTION=EMAIL,
  ROUTE=IBPERS000VNET.IBM.COM,
  NAME=C. PERSON;
CONTACT ONCALLDAY=WEEKEND,ONCALLTIME=16:00 to 24:00,
  CONNECTION=ALPHAPAGE,ROUTE=6127555,
  NAME=C. PERSON,TAPACCESS=918001234567;
```

**Usage Note:**
The contact entry supports mixed case.
Chapter 5. Event/Automation Service Definition Statements

The event/automation service serves as a gateway for event data between the Tivoli NetView for z/OS management environment, Tivoli Managed Region management environment, and SNMP Trap managers. With this gateway function, you can manage all network events from the management platform of your choice.

The event/automation service converts Tivoli NetView for z/OS alerts and messages into Tivoli Enterprise Console events before forwarding the event data to a console in the Tivoli Managed Region. As a result, all network events can be managed from a Tivoli Enterprise Console. The event/automation service also converts Tivoli NetView for z/OS alerts into SNMP Traps before forwarding the trap to an SNMP manager. SNMP traps can also be forwarded to the event/automation service and converted into alerts before forwarding the alert to the Tivoli NetView for z/OS alert receiver PPI mailbox. For more information on Tivoli Enterprise Consoles, refer to the Tivoli Enterprise Console User’s Guide.

The event/automation service definition statements are contained in the following files by default if the event/automation service is started from the IHSAEVNT job:

- **IHSAINIT** (global initialization file)
- **IHSAACFG** (alert adapter service configuration file)
- **IHSAMCFG** (message adapter service configuration file)
- **IHSAEFCFG** (event receiver service configuration file)
- **IHSATCFG** (trap-to-alert service configuration file)
- **IHSAATCF** (alert-to-trap service configuration file)

The event/automation service definition statements are contained in the following files by default if the event/automation service is started from a UNIX System Services command line:

- **/etc/netview/global_init.conf** (global initialization file)
- **/etc/netview/alert_adpt.conf** (alert adapter service configuration file)
- **/etc/netview/message_adpt.conf** (message adapter service configuration file)
- **/etc/netview/event_rcv.conf** (event receiver service configuration file)
- **/etc/netview/trap_alert.conf** (trap-to-alert service configuration file)
- **/etc/netview/alert_trap.conf** (alert-to-trap service configuration file)

These statements are system controlling constants that are read when the event/automation service is initialized.

### AdapterCdsFile

The AdapterCdsFile definition statement specifies the name of the Class Definition Statement (CDS) file. The AdapterCDSFile is used by the Alert Adapter service for converting alerts into Tivoli Enterprise Console events, by the Event Receiver service for converting Tivoli Enterprise Console events into alerts, by the Trap to Alert service for converting SNMP traps into alerts, and by the Alert to Trap service for converting alerts into SNMP traps.

The syntax for the AdapterCdsFile statement follows:
### AdapterCdsFile

Where:

The `cdsfile` variable specifies the alert adapter, event receiver, trap-to-alert, or alert-to-trap CDS file. If preceded with the backslash (`\`) character, the `cdsfile` is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

Usage Notes:

- The `AdapterCdsFile` statement is found in the alert adapter, event receiver, trap-to-alert, and alert-to-trap service configuration files.
- If you specify a complete file name, the backslash (`\`) character must immediately precede the complete filename; there can be no intervening spaces.
- If you do specify `AdapterCdsFile` for the alert adapter service, a default of `/etc/netview/alert_adpt.cds` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSAACDS is used if the event/automation service is started from the IHSAEVNT job.
- If you do not specify `AdapterCdsFile` for the event receiver service, a default of `/etc/netview/event_rcv.cds` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSAECDS is used if the event/automation service is started from the IHSAEVNT job.
- If you do not specify `AdapterCdsFile` for the trap-to-alert service, a default of `/etc/netview/trap_alert.cds` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSATCDS is used if the event/automation service is started from the IHSAEVNT job.
- If you do not specify `AdapterCdsFile` for the alert-to-trap service, a default of `/etc/netview/alert_trap.cds` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSALCDS is used if the event/automation service is started from the IHSAEVNT job.
- If the event/automation service is started from an UNIX System Services command line, use a complete file name.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

**Related Statements:** AdapterFmtFile

### AdapterFmtFile

The `AdapterFmtFile` statement specifies the name of the format (FMT) file. The `AdapterFmtFile` is used by the message adapter service for converting messages into Tivoli Enterprise Console events.

The syntax for the `AdapterFmtFile` statement follows:
Where:

The `fmtfile` variable specifies the Message Adapter FMT file. If preceded with the backslash (`\`) character, the `AdapterFmtFile` filename is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

Usage Notes:

- This statement is found in the message adapter service configuration file.
- If you specify a complete filename, the backslash (`\`) character must immediately precede the complete filename; there can be no intervening spaces.
- If you do not specify the `AdapterFmtFile` statement, a default of `/etc/netview/message_adpt.conf` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSAMFMT is used if the event/automation service is started from the IHSAEVNT job.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: `AdapterCdsFile`

---

ALRTCFG

The ALRTCFG statement specifies the name of the alert adapter service configuration file.

The syntax for the ALRTCFG statement follows:

```
ALRTCFG=
```

Where:

The `filename` variable specifies the alert adapter service configuration file. If preceded with the backslash (`\`) character, the filename is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

Usage Notes:

- This statement is found in the global initialization file.
- If you specify a complete filename, the backslash (`\`) character must immediately precede the complete filename; there can be no intervening spaces.
- If you do not specify the ALRTCFG statement, a default of `/etc/netview/alert_adpt.conf` is used if the event/automation service is started from a UNIX system services command line, or a default of IHSAMFMT is used if the event/automation service is started from the IHSAEVNT job.
**ALRTCFG**

from a UNIX System Services command line, or a default of IHSAACFG is used if the event/automation service is started from the IHSAEVNT job.

- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.
- This statement can be overridden with the ALRTCFG startup parameter. Refer to the *Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on event/automation service startup parameters.

**Related Statements:** MSGCFG, ERCVCFG, TALRTCFG, ALRTTCFG

---

**ALRTTCFG**

The ALRTTCFG statement specifies the name of the Alert to Trap service configuration file.

The syntax for the ALRTTCFG statement follows:

```
ALRTTCFG = filename
```

Where:

The `filename` variable specifies the alert to trap service configuration file. If preceded with the backslash (`\`) character, the file name is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

**Usage Notes:**

- This statement is found in the global initialization file.
- If you specify a complete file name, the backslash (`\`) character must immediately precede the complete file name; there can be no intervening spaces.
- If you do not code the ALRTTCFG statement, a default of 
  `\etc/netview/alert_trap.conf` is used if the event/automation service is started UNIX system services command line, or a default of IHSAATCF is used if the event/automation service is started from the IHSAEVNT job.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.
- This statement can be overridden with the ALRTTCFG startup parameter. Refer to the *Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on event/automation service startup parameters.

**Related Statements:** ALRTCFG, MSGCFG, ERCVCFG, TALRTCFG

---
BufEvtMaxSize

The BufEvtMaxSize statement specifies the maximum size, in kilobytes, of the Message Adapter service or Alert Adapter service event buffer file.

The syntax for the BufEvtMaxSize statement is:

```
BufEvtMaxSize= <size>
```

Where:

The `size` variable specifies the size in KB of the event buffer file.

Usage Notes:

- This statement is found in the Message Adapter service configuration file and the Alert Adapter service configuration file.
- If you do not specify the BufEvtMaxSize statement, a default of 64 is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.


BufEvtPath

The BufEvtPath statement specifies the full path name of the Message Adapter service or Alert Adapter service event buffer file.

The syntax for the BufEvtPath statement is:

```
BufEvtPath= <filename>
```

Where:

The `filename` variable specifies the full path name of the buffer event file.

Usage Notes:

- This statement is found in the message adapter service configuration file and the alert adapter service configuration file. In the sample message adapter service configuration file, it is set to `/etc/Tivoli/tec/cache_nv390msg`. In the sample alert adapter service configuration file, it is set to `/etc/Tivoli/tec/cache_nv390alt`.
- If you do not specify the BufEvtPath statement, a default of `/etc/Tivoli/tec/cache` is used.
- Only HFS files can be used for this statement; MVS data sets are not supported.

If you specify a value that is not valid, an error message is issued and the specific service terminates.
BufEvtPath

**Related Statements:** BufEvtMaxSize, BufEvtRdBlklen, BufEvtShrinkSize, BufferEvents, BufferFlushRate, BufferEventsLimit, FilterCache.

---

### BufEvtRdBlklen

The BufEvtRdBlklen definition statement specifies the size, in KB, of the buffer used by the message adapter service or alert adapter service to read data from the event buffer. Data is read from the event buffer in data blocks of the specified size.

The syntax for the BufEvtRdBlklen statement follows:

```
BufEvtRdBlklen=size
```

Where:

The `size` variable specifies the size in KB.

**Usage Notes:**

- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not specify the BufEvtRdBlklen definition statement, a default of 64 is used.
- If any single event in the event buffer is larger than the BufEvtRdBlklen, the event is discarded.
- If you specify a value that is not valid, an error message is issued and the specific service ends.

**Related Statements:** BufEvtMaxSize, BufEvtPath, BufEvtShrinkSize, BufferEvents, BufferFlushRate, BufferEventsLimit, FilterCache.

---

### BufEvtShrinkSize

The BufEvtShrinkSize definition statement specifies the amount, in KB, to shrink the message adapter service or alert adapter service event buffer file when BufEvtMaxSize is exceeded.

The syntax for the BufEvtShrinkSize statement follows:

```
BufEvtShrinkSize=amount
```

Where:

The `amount` variable specifies the amount.

**Usage Notes:**

- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not specify the BufEvtShrinkSize statement, a default of 8 is used.
If the event buffer will be shrunk, the oldest events in the buffer are discarded first until the shrink size is reached.

If you specify a value that is not valid, an error message is issued and the specific service terminates.

If the shrink size does not fall on an event boundary in the event buffer, the file is shrunk to the next whole event.

**Related Statements:** BufEvtMaxSize, BufEvtPath, BufEvtRdBlklen, BufferEvents, BufferFlushRate, BufferEventsLimit, FilterCache.

---

### BufferEvents

The BufferEvents definition statement specifies whether or not event buffering is enabled for the message adapter service or alert adapter service.

The syntax for the BufferEvents statement is:

```
BufferEvents= YES
```

Where:

- **YES** or **yes** Specifies that event buffering is enabled.
- **NO** or **no** Specifies that event buffering is disabled.

**Usage Notes:**

- This statement is found in the message adapter service configuration file and the Alert Adapter service configuration file.
- If you do not specify the BufferEvents statement, a default of YES is used.
- If you specify an incorrect value, event buffering will be disabled.
- If you specify a value that is not valid, an error message is issued and the specific service ends.

**Related Statements:** BufEvtMaxSize, BufEvtPath, BufEvtRdBlklen, BufEvtShrinkSize, BufferFlushRate, BufferEventsLimit, FilterCache.

---

### BufferEventsLimit

The BufferEventsLimit definition statement specifies the maximum number of events that can be buffered during the current connection outage for the message adapter service or alert adapter service. A value of 0 indicates that there is no limit.

The syntax for the BufferEventsLimit statement follows:

```
BufferEventsLimit= limit
```

---
**BufferEventsLimit**

Where:

The *limit* variable specifies the maximum number of events.

Usage Notes:

- Whenever an event is successfully sent to a Tivoli Enterprise Console server, the current count of events that have been buffered is reset to zero (0). This count is incremented each time an event is buffered.
- Events that are in the event buffer when the message adapter service or alert adapter service is started do not count toward the BufferEventsLimit.
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not specify the BufferEventsLimit definition statement, a default of 0 is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.


**BufferFlushRate**

The BufferFlushRate definition statement specifies the number of buffered events sent per minute by the message adapter service or alert adapter service when a lost connection has been recovered.

The syntax for the BufferFlushRate statement follows:

```
BufferFlushRate=rate
```

Where:

The *rate* variable specifies the number of events per minute.

Usage Notes:

- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not code the BufferEvents statement, a default of zero (0) is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.


**Community**

The Community statement specifies a community name that the z/OS SNMP agent is configured to support. Consult z/OS documentation for the SNMP agent for more information.

The syntax for the Community statement follows:

```
Community=community_name
```
Community

Community Name

Where:

community_name

Specifies the community name that the z/OS SNMP agent is configured to support. The community_name can be up to 32 characters in length and is case-sensitive. The public community name is the default.

Usage Notes:

- This statement is found in the alert-to-trap configuration file.
- If you do not code the community statement, a default of "public" is used.
- If you specify a value that is not valid, an error message is issued and the alert-to-trap service terminates.

ConnectionMode

The ConnectionMode statement specifies how to connect to the Tivoli Enterprise Console server for the message adapter service or alert adapter service.

The syntax for the ConnectionMode statement follows:

ConnectionMode=

Where:

connection_less specifies that a new IP socket connection is to be established for each Tivoli Enterprise Console event that is to be sent to the server.

connection_oriented or CO or co specifies that the IP socket connection should be retained between sending Tivoli Enterprise Console events to the server.

Usage Notes:

- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not code the ConnectionMode statement, a default of connection_less is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: RetryInterval.
Enterpriseoid

The Enterpriseoid statement specifies the enterprise object ID to use when building traps for the alert-to-trap service. The default is the object ID for Tivoli NetView for z/OS.

The syntax for the Enterpriseoid statement follows:

```
Enterpriseoid=1.3.6.1.4.1.1.1588.1.3
```

Where:

The `oid` variable specifies the enterprise object ID to use when building traps for the alert-to-trap service.

Usage Notes:
- This statement is found in the alert-to-trap configuration file.
- If you do not code the Enterprise oid, a default of 1.3.6.1.4.1.1.1588.1.3 is used.
- If you specify a value that is not valid, an error message is issued and the alert-to-trap service terminates.

ERCVCFG

The ERCVCFG statement specifies the name of the Event Receiver service configuration file.

The syntax for the ERCVCFG statement follows:

```
ERCVCFG=IHSAECFG\etc\netview\event_rcv.conf
```

Where:

The `filename` variable specifies the event receiver service configuration file. If preceded with the backslash (`\`) character, the filename is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

Usage Notes:
- This statement is found in the global initialization file.
- If you specify a complete file name, the backslash (`\`) character must immediately precede the complete file name; there can be no intervening spaces.
- If you do not code the ERCVCFG statement, a default of \etc\netview\event_rcv.conf is used if the event/automation service is started from a UNIX system services command line, or a default of IHSAECFG is used if the event/automation service is started from the IHSAEVNT job.
• If you specify a value that is not valid, an error message is issued and the event/automation service terminates.

• This statement can be overridden with the ERCVCFG startup parameter. Refer to the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information on event/automation service startup parameters.

Related Statements: ALRTCFCFG, MSGCFG, TALRTCFCFG, ALRTTCFCFG

EventMaxSize

The EventMaxSize statement specifies the maximum length of the Tivoli Enterprise Console events generated by the message adapter service or alert adapter service.

The syntax for the EventMaxSize statement follows:

```
EventMaxSize=size
```

Where:

The `size` variable specifies the maximum size of an event in bytes.

Usage Notes:

• This statement is found in the message adapter service configuration file and the alert adapter service configuration file.

• If you do not code the EventMaxSize statement, a default of 4096 is used.

• If you specify a value that is not valid, an error message is issued and the specific service terminates.

Filter

The filter statement specifies the criteria used to filter events for the message adapter service or alert adapter service. The filter can be used to either pass events to the Tivoli Enterprise Console server, or to prevent events from being sent to the console server, depending on the FilterMode statement.

The syntax for the filter statement follows:

```
Filter:Class=classname; slot=value;
```

Where:

The `classname` variable specifies the name of the class for the event.

The `slot` variable specifies a slot within the event.

The `value` variable specifies the value of a particular slot within the event.
Filter

Usage Notes:
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- A Filter statement is matched if the incoming event has class and slot/value pairs that match the criteria specified in the Filter statement. The match does not have to be an exact match. The Filter statement represents a subset of the event. Once a match is made, the FilterMode statement indicates whether the matched event is to be sent or discarded.
- Multiple filter statements can be specified in the configuration file. An attempt to match on each statement is made until a match occurs or all filter statements are exhausted.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: FilterMode, FilterCache

FilterCache

The FilterCache statement specifies the criteria used to filter events for event buffering for the message adapter service or alert adapter service. The filter can be used to either pass events to the event buffer, or to prevent events from being buffered, depending on the FilterMode statement.

The syntax for the FilterCache statement follows:

```
FilterCache:Class=classname; slot=value;
```

Where:

The `classname` variable specifies the name of the class for the event.

The `slot` variable specifies a slot within the event.

The `value` variable specifies the value of a particular slot within the event.

Usage Notes:
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- Tivoli Enterprise Console events that are discarded due to a filter statement match are not passed through the FilterCache statements.
- A FilterCache statement is matched if the incoming event has class and slot/value pairs that match the criteria specified in the filter statement. The match does not have to be an exact match. The FilterCache statement represents a subset of the event. Any event that matches a FilterCache statement will not be buffered.
- Multiple FilterCache statements can be specified in the configuration file. An attempt to match on each statement will be made until a match is made or all FilterCache statements are exhausted.
FilterMode

The FilterMode statement specifies how events that match a Filter or FilterCache statement are to be treated. These events can either be sent or discarded for Filter statements, or buffered or not buffered for FilterCache statements. This statement is used for the Message Adapter service or Alert Adapter service.

The syntax for the FilterMode statement follows:

\[ \text{FilterMode} = \text{OUT} \text{ or } \text{OUT} \]
\[ \text{out} \text{ or } \text{IN} \]
\[ \text{in} \]

Where:

**OUT** or **out** specifies that events that match a filter statement are to be discarded, and events that match a FilterCache statement are not to be buffered.

**IN** or **in** specifies that events that match a filter statement are to be sent, and events that match a FilterCache statement are to be buffered.

Usage Notes:
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not code the FilterMode statement, a default of **OUT** is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: Filter, FilterCache, BufferEvents

Hostname

The Hostname statement specifies the host name of the TCP/IP stack that is providing the z/OS SNMP agent for the alert-to-trap service. In most cases, this is the local host where the event/automation service is running. In this case, the default of **loopback** is adequate. Consult z/OS documentation for TCP/IP for further information.

The syntax for the Hostname statement follows:

\[ \text{Hostname} = \text{loopback} \text{ or } \text{hostname} \]

Where:
TALRTCFG

The *hostname* variable specifies the name of the TCP/IP stack providing the SNMP agent.

**Usage Notes:**
- This statement is found in the alert-to-trap configuration file.
- If you do not code the hostname statement, a default of `loopack` is used.
- If you specify a value that is not valid, an error message is issued and the alert-to-trap service terminates.

MSGCFG

The MSGCFG statement specifies the name of the message adapter service configuration file.

The syntax for the MSGCFG statement follows:

```
MSGCFG=
```

Where:

The *filename* variable specifies the message adapter service configuration file. If preceded with the backslash (`\`) character, the file name is a complete MVS data set or an HFS file name. If not preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

**Usage Notes:**
- This statement is found in the global initialization file.
- If you specify a complete file name, the backslash (`\`) character must immediately precede the complete file name; there can be no intervening spaces.
- If you do not code the MSGCFG statement, a default of `\etc/netview/alert_adpt.conf` is used if the event/automation service is started from a UNIX System Services command line, or a default of `IHSAMCFG` is used if the event/automation service is started from the IHSAEVNT job.
- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.
- **This statement can be overridden with the MSGCFG startup parameter.** Refer to the *Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on event/automation service startup parameters.

**Related Statements:** ALRTCFG, ERCVCFG, TALRTCFG, ALRTTCFG
NetViewAlertReceiver

The NetViewAlertReceiver statement specifies the PPI mailbox name that identifies the NetView alert receiver program.

The syntax for the NetViewAlertReceiver statement follows:

```
NetViewAlertReceiver= NETVALRT ppiname
```

Where:

The `ppiname` variable specifies a 1–8 character PPI mailbox name for the NetView alert receiver program.

Usage Notes:

- This statement is found in the event receiver and trap-to-alert service configuration file.
- If you do not code the NetViewAlertReceiver statement, a default of NETVALRT is used.
- If you specify a value that is not valid, an error message is issued and the event receiver service terminates.

NOSTART

The NOSTART statement specifies which service tasks, if any, are not to be started when the event/automation service is initialized.

The syntax for the NOSTART statement follows:

```
NOSTART TASK= EVENTRCV MESSAGEA ALERTA TRAPALRT ALRTTRAP ALL
```

Where:

The `EVENTRCV` variable specifies that the event receiver service is not to be started.
**NOSTART**

The `MESSAGEA` variable specifies that the message adapter service is not to be started.

The `ALERTA` variable specifies that the alert adapter service is not to be started.

The `TRAPALRT` variable specifies that the trap-to-alert service is not to be started.

The `ALRTTRAP` variable specifies that the alert-to-trap service is not to be started.

The `ALL` variable specifies that all services are not to be started.

**Usage Notes:**

- This statement is found in the global initialization file.
- If you do not code the NOSTART statement, all services are started (this statement does not have a default setting).
- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.

---

**OUTPUT**

The OUTPUT statement specifies the logical destination of trace/error data. Trace/error data can be sent to the system output log file, the GTF trace facility, or to both.

The syntax for the OUTPUT statement follows:

```
OUTPUT=(SYSOUT),(GTF),(ALL)
```

Where:

The `SYSOUT` variable specifies that trace/error data is to be sent to the system output file. Each task within the event/automation service has a system output file.

`GTF` specifies that trace/error data is to be sent to the generalized trace facility (GTF) if active.

`ALL` specifies that trace/error data is to be sent to both the GTF and the system output file.

**Usage Notes:**

- This statement is found in the global initialization file.
- If you do not code the OUTPUT statement, a default of SYSOUT is used.
- The GTF can be activated at anytime. If GTF is not active when an attempt is made by the event/automation service to send trace/error data to the GTF, a
single warning message will be issued. When the GTF is activated, the event/automation service will begin sending trace/error data to it. Any data sent to GTF while it is inactive will be lost, unless it is also being sent to the system output file.

- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.
- This statement can be overridden with the OUTPUT modification command.
  Refer to the Tivoli NetView for z/OS Command Reference or the NetView online help for more information about event/automation service commands.

**PortNumber**

The PortNumber statement specifies the port number that the event receiver and trap-to-alert services will use.

The syntax for the PortNumber statement follows:

```
PortNumber= port
```

**Where:**

The `port` variable specifies a valid IP port number. If zero (0), this indicates that the service should use PortMapper to assign a port.

**Usage Notes:**

- This statement is found in the event receiver and trap-to-alert service configuration file.
- If you do not code the PortNumber statement, a default of zero (0) is used.
- If PortNumber is zero (0), the UsePortMapper statement must be set to YES for the event receiver service; otherwise an error message is issued and the event receiver service terminates.
- If you specify a value that is not valid, an error message is issued and the event receiver service terminates.

**Related Statements:** UsePortmapper.

**PPI**

The PPI statement specifies the PPI mailbox name that identifies this event/automation service.

```
PPI= ppiname
```

**Where:**

The `ppiname` variable specifies a 1–8 character PPI mailbox name for the event/automation service.
PPI

Usage Notes:
- This statement is found in the global initialization file.
- If you do not code the PPI statement, a default of IHSATEC is used.
- If you start more than one event/automation service on the same MVS system, make sure that you specify a unique PPI mailbox name for each.
- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.
- This statement can be overridden with the PPI startup parameter. Refer to the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information on event/automation service startup parameters.

RetryInterval

The RetryInterval statement specifies how many seconds the message adapter service or alert adapter service is to wait before attempting to reestablish a TCP/IP session with a Tivoli Enterprise Console server. This statement only has meaning if the ConnectionMode is connection_oriented.

The syntax for the RetryInterval statement follows:

```plaintext
RetryInterval=interval
```

Where:

The `interval` variable specifies the number of seconds to wait before an attempt to reestablish the connection is made.

Usage Notes:
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not code the RetryInterval statement, a default of 120 is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: ConnectionMode

ServerLocation

The ServerLocation statement specifies the name of the host on which the Tivoli Enterprise Console server is installed. This is used by the message adapter and alert adapter services for forwarding converted console events.

The syntax for the ServerLocation statement follows:

```plaintext
ServerLocation=servername
```

Where:
The `servername` variable specifies the location of the Tivoli Enterprise Console event server, and must be one of the following formats:

- A TCP/IP host name or TCP/IP address. This form is used if the adapters are using a non-secure connection to the Tivoli Enterprise Console server.
- `@EventServer`, where `EventServer` is the name of the Tivoli Enterprise Console server. This form is used if the adapters are using a secure connection to the Tivoli Enterprise Console server through the Client Framework and the Tivoli Enterprise Console server is in the local management region.
- `@EventServer#RegionName`, where `EventServer` is the name of the Tivoli Enterprise Console server and `RegionName` is the name of the TMR region where the Tivoli Enterprise Console server resides. This form is used if the adapters are using a secure connection to the Tivoli Enterprise Console server through the Client Framework and the console server is in another management region.
- An HFS fully qualified filename if the TestMode statement indicates that Tivoli Enterprise Console events are to be sent to a file for debug purposes.

**Usage Notes:**

- This statement is found in the alert adapter service configuration file and the message adapter service configuration file.
- The `ServerLocation` statement must be coded.
- `ServerLocation` can contain up to 16 comma-separated values. The first location is the primary Tivoli Enterprise Console server, the other locations are secondary console servers that are to be used only when a connection cannot be established to the primary console server.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

**Related Statements:** `ServerPort`

---

**ServerPort**

The `ServerPort` statement specifies the port numbers on which the hosts specified in the `ServerLocation` listen. This is used by the message adapter and alert adapter services for forwarding converted Tivoli Enterprise Console events.

The syntax for the `ServerPort` statement follows:

```
ServerPort=port, port
```

**Where:**

The `port` variable specifies the port numbers.

**Usage Notes:**

- This statement is found in the alert adapter service configuration file and the message adapter service configuration file.
ServerPort

- If you do not code the ServerLocation statement, its value is zero (0). A value of zero (0) indicates that PortMapper should be used to retrieve the port numbers of the corresponding Tivoli Enterprise Console server.

- If you do not code the ServerLocation statement, its value is (0). A value of zero (0) indicates that PortMapper should be used to retrieve the port numbers of the corresponding Tivoli Enterprise Console server.

- ServerPort contains up to 16 comma-separated values. The first port corresponds to the first host name listed in the ServerLocation statement, the second port to the second host name, and so on.

- If you specify a value that is not valid, an error message is issued and the specific service terminates.

Related Statements: ServerLocation

TALRTCFG

The TALRTCFG statement specifies the name of the Trap to Alert service configuration file.

The syntax for the TALRTCFG statement follows:

```
TALRTCFG=
IHSATCFG
\etc/netview/trap_alert.conf
filename
```

Where:

The `filename` variable specifies the trap-to-alert service configuration file. If preceded with the backslash (`\`) character, the file name is a 1–8 character member name associated with the IHSSMP3 data set definition from the IHSAEVNT procedure.

Usage Notes:

- This statement is found in the global initialization file.

- If you specify a complete file name, the backslash (`\`) character must immediately precede the complete file name; there can be no intervening spaces.

- If you do not code the TALRTCFG statement, a default of `\etc/netview/trap_alert.conf` is used if the event/automation service is started UNIX system services command line, or a default of IHSATCFG is used if the event/automation service is started from the IHSAEVNT job.

- If you specify a value that is not valid, an error message is issued and the specific service terminates.

- This statement can be overridden with the TALRTCFG startup parameter. Refer to the Tivoli NetView for z/OS Installation: Configuring Additional Components for more information on event/automation service startup parameters.

Related Statements: ALRTCFG, MSGCFG, ERCVCFG, ALRTTCFG
The TestMode statement specifies whether Tivoli Enterprise Console events are to be sent to the console server or sent to a debug file for the message adapter service or alert adapter service. The debug file is specified in the ServerLocation statement if the test mode is enabled.

The syntax for the TestMode statement follows:

```
TestMode = NO
      YES
      yes
      NO
      no
```

Where:

- The YES or yes variable specifies that the test mode is enabled. All events are routed to the debug file.
- NO or no specifies that the test mode is disabled.

Usage Notes:
- This statement is found in the message adapter service configuration file and the alert adapter service configuration file.
- If you do not code the TestMode statement, a default of NO is used.
- If you specify a value that is not valid, an error message is issued and the specific service terminates.

The TRACE statement specifies what level of tracing to enable for one or more service tasks.

The syntax for the TRACE statement follows:

```
TRACE TASK = 
   EVENTRCV
   MESSAGEA
   ALERTA
   TRAPALRT
   ALRTRAP
   CONTROL
   ALL

   LEVEL = OFF
      LOW
      NORMAL
      VERBOSE

```

Chapter 5. Event/Automation Service Definition Statements 325
Where:

The EVENTRCV variable specifies that the event receiver task is to be traced.

The MESSAGEA variable specifies that the message adapter task is to be traced.

The ALERTA variable specifies that the alert adapter task is to be traced.

The TRAPALRT variable specifies that the trap-to-alert service is to be traced.

The ALRTTRAP variable specifies that the alert-to-trap service is to be traced.

The CONTROL variable specifies that the control task is to be traced.

The ALL variable specifies that all tasks are to be traced.

The OFF variable specifies that tracing for the specified tasks is disabled.

The LOW variable specifies that the lowest level of tracing is enabled. Typically, this traces functions entry and exit.

The NORMAL variable specifies the typical level of tracing is enabled. This traces the LOW level, plus any function flow information that aids problem determination.

The VERBOSE variable specifies the highest level of tracing is enabled. This traces the NORMAL level, plus any pertinent control blocks that aid problem determination. This level of tracing can result in a large amount of data being sent to the trace/error log and/or GTF.

The IP variable specifies that the tracing of IP connection data should be enabled or disabled.

Usage Notes:

- This statement is found in the global initialization file.
- If you specify a value that is not valid, an error message is issued and the event/automation service terminates.
- If you do not code the TRACE statement, all tracing is disabled for all tasks.
- This statement can be overridden with the TRACE modification command. Refer to the Tivoli NetView for z/OS Command Reference for more information on event/automation service commands.

**UsePortMapper**

The UsePortMapper statement specifies the whether or not the event receiver service will register itself with the PortMapper program. If the port number supplied on the PortNumber statement is zero (0), PortMapper will also be used to assign a port to the event receiver service.
The syntax for the UsePortmapper statement follows:

```
UsePortMapper=

YES

yes

NO

no
```

Where:

The **YES** or **yes** variable specifies the PortMapper is to be used.

The **NO** or **no** variable specifies that PortMapper is not to be used.

Usage Notes:

- This statement is found in the event receiver service configuration file.
- If you do not code the UsePortmapper statement, a default of YES is used.
- If UsePortmapper is NO, the PortNumber statement must have a nonzero value.
- If you specify a value that is not valid, an error message is issued and the event receiver service terminates.

Related Statements: PortNumber
UsePortMapper
Chapter 6. Resource Object Data Manager Definition
Statements

This chapter describes the statements you can use to tune the Resource Object Data Manager (RODM) data cache. The sample statements included with the NetView program are in the EKGCUST member, which is in the CNMSAMP data set.

ASYNC_TASKS

The ASYNC_TASKS statement enables you to define the number of asynchronous tasks that can run at the same time. It also controls the multiprogramming level of RODM and asynchronous method application programming interface (API) tasks.

The syntax for the ASYNC_TASKS statement is:

ASYNC_TASKS

\[
ASYNC\_TASKS(5, nn)\]

Where:

5

Specifies the number of concurrently running asynchronous tasks. You can specify a value in the range of 1-50. The default is 5.

Usage Notes:

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

CELL_POOLS

The CELL_POOLS statement enables you to define the number and size of cell pools. Storage is allocated from the operating system in large blocks (cell pools) and is subdivided by RODM. The cell pools are divided into multiple elements of the same size. You can specify up to 200 CELL_POOLS statements to identify cell size and pool size combinations.

The syntax for the CELL_POOLS statement is:

CELL_POOLS

\[
CELL\_POOLS(cellsize:poolsize)\]
**CELL_POOLS**

Where:

- **cellsize**
  Specifies the cell size of each pool, in bytes. You can specify a value in the range of 8–32K bytes. If you duplicate a cell size that RODM has already defined, RODM uses the greater of the two pool sizes.

- **poolsize**
  Specifies the number of 4K pages allocated for the pool. You can specify a value in the range of 4K–128K bytes. The default is 4K. Specify poolsize in terms of 4K memory pages, where one CELL_POOL equals 4K bytes of memory. When you specify CELL_POOLS, RODM calculates a maximum pool size for poolsize. The value (poolsize*4K) must be less than the calculated maximum poolsize; otherwise the maximum poolsize is used. The default value for poolsize is 1. If the cellsize is less than 8 bytes, it is adjusted to 8 bytes. If the poolsize*4K is less than the cellsize, the poolsize is adjusted to a value that provides the minimum storage for one cell.

Usage Notes:
- It is not necessary to specify a CELL_POOLS value. If you specify CELL_POOLS ( ), the following 29 defaults for CELL_POOLS are used:

<table>
<thead>
<tr>
<th>Cell Size (bytes)</th>
<th>Pool Size (pages)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>40</td>
<td>2</td>
</tr>
<tr>
<td>48</td>
<td>2</td>
</tr>
<tr>
<td>52</td>
<td>2</td>
</tr>
<tr>
<td>56</td>
<td>2</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>68</td>
<td>3</td>
</tr>
<tr>
<td>72</td>
<td>3</td>
</tr>
<tr>
<td>80</td>
<td>3</td>
</tr>
<tr>
<td>88</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>3</td>
</tr>
<tr>
<td>104</td>
<td>3</td>
</tr>
<tr>
<td>112</td>
<td>3</td>
</tr>
<tr>
<td>120</td>
<td>3</td>
</tr>
<tr>
<td>128</td>
<td>3</td>
</tr>
<tr>
<td>136</td>
<td>4</td>
</tr>
<tr>
<td>144</td>
<td>4</td>
</tr>
<tr>
<td>152</td>
<td>4</td>
</tr>
<tr>
<td>160</td>
<td>4</td>
</tr>
</tbody>
</table>

  - You can specify up to 200 different CELL_POOLS specifications in EKGCUST to identify different cell size and pool size combinations.
  - When you add or change CELL_POOLS statements, cold-start RODM to activate the changes.
Changes to this statement are not used in a warm start. If you warm-start RODM, RODM uses the original keyword values specified when you cold-started RODM.

This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Example:
The following is an example of CELL_POOLS statement entries in the EKGCUST customization file:

```
CELL_POOLS ( 8, 12, 16, 20 )
CELL_POOLS ( 24, 28, 32, 36:2 )
CELL_POOLS ( 40:2, 48:2, 52:2, 56:2 )
CELL_POOLS ( 60:2, 64:2, 68:3, 72:3 )
CELL_POOLS ( 80:3, 88:3, 100:3, 104:3 )
CELL_POOLS ( 112:3, 120:3, 128:3, 136:4 )
CELL_POOLS ( 144:4, 152:4, 160:4, 168:4 )
CELL_POOLS ( 176:4, 184:4, 192:4, 200:4 )
CELL_POOLS ( 208:4, 216:4, 224:4, 232:4 )
CELL_POOLS ( 240:4, 248:4, 256:4, 284:6 )
CELL_POOLS ( 484:9, 512:8, 768:9, 1024:8 )
CELL_POOLS ( 1536:12, 2048:16, 3072:24, 4096:32 )
CELL_POOLS ( 6144:48, 8192:64, 12288:96, 16384:128 )
CELL_POOLS ( 24576:192, 32768:256 )
```

### CHARACTER_VALIDATION

The CHARACTER_VALIDATION statement enables you to control the amount of validity checking RODM does for class, object, and field names.

The syntax for the CHARACTER_VALIDATION statement is:

```
CHARACTER_VALIDATION

Where:

**YES**

For class and field names, specifies that valid names can have the following characters:

- The first character of the string must be alphabetic or numeric.
- The other characters, if any, may be alphabetic, numeric, the break character (_), the commercial “at” sign (@), the number sign (#), or the period (.)

For object names, specifies that valid names can have the following characters:

- The first character of the string must be alphabetic or numeric.
- The other characters, if any, can be alphabetic, numeric, or any of the special characters: # @ . , ; ? ( ) \'' _ & plus; + * = < > /

For a complete description of valid class, object, and field names, refer to the *Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide*.
For class, object, and field names, the following restrictions apply:

- The first character cannot be the number sign (#).
- Blank characters are not allowed.
- Null characters are not allowed.

For a complete description of valid class, object, and field names, refer to the Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer’s Guide.

Usage Notes:

- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.

CHECKPOINT_FUNCTION

The CHECKPOINT_FUNCTION statement enables you to control the RODM checkpoint function.

The syntax for the CHECKPOINT_FUNCTION statement is:

```
CHECKPOINT_FUNCTION

REQUEST

NONE

REQUIRE
```

Where:

REQUEST

Specifies that the RODM checkpoint function is "requested to be enabled". With this setting, RODM will continue to allocate storage for new objects and classes, even if that storage exceeds the capacity of the checkpoint data sets. If the storage allocated by RODM grows beyond the capacity of the checkpoint data sets, the checkpoint function is disabled and RODM continues.

Additionally, this setting specifies that during a cold start initialization, RODM should continue even if a checkpoint function setup error occurs.

NONE

Specifies that the RODM checkpoint function is disabled. This option is not valid on a warm start. During a warm start, this option is flagged as an error, and a WTOR is issued, giving you the option to terminate RODM, or continue. If you choose to continue with initialization, the default (CHECKPOINT=REQUEST) is used.

REQUIRE

Specifies that the RODM checkpoint function is required. With this setting, RODM does NOT continue to allocate storage for new objects and classes if that storage would exceed the capacity of the checkpoint data sets. This causes the capacity of RODM to be limited to the capacity of the checkpoint data sets. This is how RODM operated in releases prior to TME® 10 NetView for OS/390 V1R2.
Additionally, this option specifies that during a cold start initialization, RODM should terminate if a checkpoint function setup failure occurs.

**Usage Notes:**
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

**CONCURRENT_USERS**

The CONCURRENT_USERS statement enables you to define the number of concurrent active transactions occurring in the RODM address space. This value does not limit the number of active users that can be concurrently connected to RODM.

The syntax for the CONCURRENT_USERS statement is:

```plaintext
CONCURRENT_USERS
```

Where:
- \( \text{10} \ |	ext{nnn} \)
  - Specifies the number of concurrent active transactions. You can specify a value in the range of 1–200 transactions. The default is 10.

**Usage Notes:**
- This number should reflect the maximum number of transactions you expect to occur in RODM at one time. RODM allocates storage for each transaction specified in this statement. Transactions include the number of users plus the total of asynchronous tasks running at the same time.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

**CONNECT_VIOLATION_MESSAGE**

The CONNECT_VIOLATION_MESSAGE statement enables you to specify whether RODM will write message EKG2000I when it encounters a problem while processing a CONNECT request.

The syntax for the CONNECT_VIOLATION_MESSAGE statement is:

```plaintext
CONNECT_VIOLATION_MESSAGE
```

Where:
- \( \text{NO} \ |	ext{YES} \)
  - Specifies whether RODM will write message EKG2000I when it encounters a problem while processing a CONNECT request.
CONNECT_VIOLATION_MESSAGE

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
</table>
| Specifies whether RODM writes message EKG2000I to the system log when it encounters a problem while processing a CONNECT request. Specify YES or NO. The default is NO.

DUMP_FOR_BAD_USER_DATA

The DUMP_FOR_BAD_USER_DATA statement enables you to specify whether RODM will dump when it encounters a problem while copying user data.

The syntax for the DUMP_FOR_BAD_USER_DATA statement is:

```
DUMP_FOR_BAD_USER_DATA
```

Where:

<table>
<thead>
<tr>
<th>NO</th>
<th>YES</th>
</tr>
</thead>
</table>
| Specifies whether the RODM will dump when it encounters a problem while copying user data. You can specify YES or NO. The default is NO.

Related Statements: DUMP_LIMIT, DUMP_SCOPE

DUMP_LIMIT

The DUMP_LIMIT statement enables you to control the rate at which dumps for RODM are taken. This can help ensure that a recurring error does not schedule more dumps than can be handled by the available system resources. This option is specified with two values, which indicate the maximum rate at which dumps are taken.

The syntax for the DUMP_LIMIT statement is:

```
DUMP_LIMIT
```

Where:

<table>
<thead>
<tr>
<th>10</th>
<th>NUMBER_OF_DUMPS</th>
</tr>
</thead>
</table>
| Specifies the number of dumps to be taken for RODM within the time interval specified by time_interval. You can specify a value from 1–30. The default is 10.

<table>
<thead>
<tr>
<th>1440</th>
<th>TIME_INTERVAL</th>
</tr>
</thead>
</table>
| Specifies the number of minutes that RODM uses as a time interval for allowing the number of dumps specified by NUMBER_OF_DUMPS.

The time interval is a sliding window. The number of dumps allowed is compared against the number of dumps taken during the previous time...
interval. For example, if \texttt{TIME\_INTERVAL} is 60 minutes, another dump is possible if \texttt{NUMBER\_OF\_DUMPS} has not already been reached during the last hour.

If you do not want to limit the number of dumps, set \texttt{NUMBER\_OF\_DUMPS} to the maximum value (30), and \texttt{TIME\_INTERVAL} to the minimum value (1). You can specify a value from 1–40320. The default is 1440 (one day).

\textbf{Usage Notes:}

- When \texttt{NUMBER\_OF\_DUMPS} is reached for a time interval, message EKG2305I is issued to warn the operator that no more RODM dumps are taken until the interval passes.
- The MVS MODIFY command can clear the dump statistics used to enforce the \texttt{DUMP\_LIMIT}. Refer to the DMPRESET command for more information.
- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.
- Changes to this statement are used in a warm start. If you warm start RODM, RODM uses the keyword values you specified in a customization file.

\textbf{Related Statements:} \texttt{DUMP\_FOR\_BAD\_USER\_DATA}, \texttt{DUMP\_SCOPE}

\section*{DUMP\_SCOPE}

The \texttt{DUMP\_SCOPE} statement enables you to specify which user address spaces are dumped when an ABEND occurs in RODM.

The syntax for the \texttt{DUMP\_SCOPE} statement is:

\begin{verbatim}
DUMP_SCOPE
\end{verbatim}

\textbf{Where:}

- **REQUESTOR**
  Specify \texttt{REQUESTOR} to dump the address space that is involved in the ABEND. This is the default.

- **CONNECTED**
  Specify \texttt{CONNECTED} to dump up to 10 address spaces that are connected to RODM, in addition to the requesting address space, and the dataspaces.

Selecting this option can help ensure that complete documentation is recorded when RODM encounters an error.

If more than 10 address spaces are connected to RODM at the time of the error, the address spaces that were connected first are dumped. If the requesting address space is one of those 10 connected address spaces, only 9 other address spaces will be dumped.

\textbf{Usage Notes:}
DUMP_SCOPE

- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.
- Changes to this statement are used in a warm start. If you warm start RODM, RODM uses the keyword values you specified in a customization file.

Related Statements: DUMP_FOR_BAD_USER_DATA, DUMP_SCOPE, DUMP_LIMIT

EXTEND_HEAP_SIZE

The EXTEND_HEAP_SIZE statement enables you to define the size, in bytes, of the extended PL/I heap storage.

The syntax for the EXTEND_HEAP_SIZE statement is:

EXTEND_HEAP_SIZE

Where:

- 32K | nnnK
  Specifies the size of the extended PL/I heap storage. You can specify a value in the range of 4 K–64 K that is a multiple of 4 K. The default is 32 K.

Usage Notes:
- This statement is invoked only if RODM exhausts the primary allocation of PL/I heap storage.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: PLI_Isa, PRIMARY_HEAP_SIZE

IO_QUEUE_THRESHOLD

The IO_QUEUE_THRESHOLD statement enables you to define the number of log requests to hold in the I/O request queue.

The syntax for the IO_QUEUE_THRESHOLD statement is:

IO_QUEUE_THRESHOLD

Where:
IO_QUEUE_THRESHOLD

$\text{nnnn}$

Specifies the number of log requests to hold in the I/O request queue. You can specify a value in the range of 0–32767. The default is 5.

Usage Notes:
- When the number of log requests reaches the value specified on IO_QUEUE_THRESHOLD, the I/O task is posted to process the requests. You can delay the I/O process by specifying a large value.
- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.

LOG_LEVEL

The LOG_LEVEL statement enables you to define the API log level. An API return code greater than or equal to the LOG_LEVEL causes a log record to be written to the RODM log.

The syntax for the LOG_LEVEL statement is:

```
LOG_LEVEL
```

Where:

$\text{8nnn}$

Specifies the API log level. You can specify a value from 0–999. The default is 8.

Usage Notes:
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.
- Using an API log level of zero (0) logs all RODM API requests. An shortage of auxiliary storage can occur in a high-stress environment.

Related Statements: MLOG_LEVEL

MAX_CHUNK

The MAX_CHUNK statement enables you to define the maximum number of free chunks of storage kept for RODM methods. This statement also enables you to control the amount of free short-lived storage allowed to accumulate before it is freed.

The syntax for the MAX_CHUNK statement is:
**MAX_CHUNK**

MAX_CHUNK

| 256 | nnnn |

Where:

256 | nnnn

Specifies the number of chunks of free storage kept. You can specify a value in the range of 16–4096. The default is 256.

Usage Notes:

- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.

Related Statements: None

**MAX_SEGMENT_NUM**

The MAX_SEGMENT_NUM statement enables you to define the maximum number of available segments allowed in the translation window. The size of each segment is 16MB.

The syntax for the MAX_SEGMENT_NUM statement is:

MAX_SEGMENT_NUM

| 64 | nnn |

Where:

64 | nnn

Specifies the maximum number of segments. You can specify a value in the range of 3–64. The default is 64.

Usage Notes:

- The MAX_SEGMENT_NUM and the MAX_WINDOW_NUM determine how much data can be stored in RODM.
- If you define or change the MAX_SEGMENT_NUM statement, cold-start RODM to activate the changes.
- Changes to this statement are not used in a warm start. If you warm-start RODM, RODM uses the original keyword values specified when you cold-started RODM.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.
Related Statements: MAX_WINDOW_NUM, SEGMENT_POCKETS, WINDOW_POCKETS

MAX_WINDOW_NUM

The MAX_WINDOW_NUM statement enables you to define the maximum number of available windows in the data spaces. The size of each window is 16MB.

The syntax for the MAX_WINDOW_NUM statement is:

```
MAX_WINDOW_NUM ( \[191\]  \\
nnn)
```

Where:

\[191\]nnn

- Specifies the maximum number of windows. You can specify a value in the range of 4–191. The default is 191.

Usage Notes:

- The MAX_SEGMENT_NUM and the MAX_WINDOW_NUM determine how much data can be stored in RODM.
- If you define or change the MAX_WINDOW_NUM statement, cold-start RODM to activate the changes.
- Changes to this statement are not used in a warm start. If you warm-start RODM, RODM uses the original keyword values specified when you cold-started RODM.

Related Statements: SEGMENT_POCKETS, WINDOW_POCKETS

MLOG_LEVEL

The MLOG_LEVEL statement enables you to define the method application programming interface (MAPI) log level. An MAPI return code greater than or equal to the MLOG_LEVEL causes a log record to be written to the RODM log.

The syntax for the MLOG_LEVEL statement is:

```
MLOG_LEVEL ( \[8\]  \\
nnn)
```

Where:

\[8\]nnn

- Specifies the MAPI log level. You can specify a value in the range of 0–999. The default is 8.
MLOG_LEVEL

Usage Notes:

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: LOG_LEVEL

MTRACE_TYPE

The MTRACE_TYPE statement enables you to define the bit map that specifies enabling and disabling of method tracing.

The syntax for the MTRACE_TYPE statement is:

```
MTRACE_TYPE
```

Where:

```
X'000000FC'
```

Specifies the bit map. You can only specify a value in the range of X'00000000'–X'000000FF' because the rest of the bits are reserved. The default is X'000000FC'.

Usage Notes:

- The bit map also specifies the types of methods to be traced for a user as follows:

<table>
<thead>
<tr>
<th>Value</th>
<th>Method Traced</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'00000001'</td>
<td>Method entry</td>
</tr>
<tr>
<td>X'00000002'</td>
<td>Method exit</td>
</tr>
<tr>
<td>X'00000004'</td>
<td>QUERY method</td>
</tr>
<tr>
<td>X'00000008'</td>
<td>CHANGE method</td>
</tr>
<tr>
<td>X'00000010'</td>
<td>NOTIFY method</td>
</tr>
<tr>
<td>X'00000020'</td>
<td>NAMED method</td>
</tr>
<tr>
<td>X'00000040'</td>
<td>OI method</td>
</tr>
<tr>
<td>X'00000080'</td>
<td>Object-deletion method</td>
</tr>
</tbody>
</table>

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

PLI_IsA

The PLI_IsA statement enables you to define the size of the PL/I internal storage area (ISA), in bytes.

The syntax for the PLI_IsA statement is:
PLI ISA

Where:

\[ 40K \leq nnK \]

Specifies the size of the PL/I internal storage area, in bytes. You can specify a value in the range of 4K–256K that is a multiple of 4K. The default is 40K.

Usage Notes:

- If your allocation of PL/I internal storage is too low, there is an increase in MVS storage requests. However, assigning too much storage reduces storage requests at the expense of virtual storage.
- All processing within the RODM address space is performed using PL/I run-time storage management enabling methods. These methods are to be written in PL/I V2R3 or IBM program product C/370™ V2R1 or later release. For efficiency, RODM pre-allocates a number of PL/I environments. The number of pre-allocated environments, as well as their size, is determined by the CONCURRENT_USERS and ASYNC_TASKS statements.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: ASYNC_TASKS, CONCURRENT_USERS, EXTEND_HEAP_SIZE, PRIMARY_HEAP_SIZE

**PRIMARY_HEAP_SIZE**

The PRIMARY_HEAP_SIZE statement enables you to define the size, in bytes, of the primary PL/I heap storage for each user thread.

The syntax for the PRIMARY_HEAP_SIZE statement is:

\[
\text{PRIMARY_HEAP_SIZE} \quad (64K \leq nnK)
\]

Where:

\[ 64K \leq nnK \]

Specifies the size of the primary PL/I heap storage, in bytes. You can specify a value in the range of 4K–256K that is a multiple of 4K. The default is 64K.

Usage Notes:

- If you do not specify adequate primary heap storage and it is exhausted, RODM makes repeated requests for extended heap storage.
**Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.**

**This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.**

**Related Statements:** PLIlsa, EXTEND_HEAP_SIZE

---

**QUIES_WAIT_TASK_TIME**

The QUIES_WAIT_TASK_TIME statement enables you to define the time interval, in increments of 0.01 seconds, during a quiesce that RODM allows for a transaction to complete before RODM prompts the user and asks if the transaction should be canceled.

The syntax for the QUIES_WAIT_TASK_TIME statement is:

```
QUIES_WAIT_TASK_TIME
```

Where:

```
200
```

Specifies the time interval, in increments of 0.01 seconds, that RODM allows for a transaction to complete during a quiesce before RODM prompts the user. You can specify a value in the range of 1–32767. The default is 200.

**Usage Notes:**

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.

**Related Statements:** QUIES_WAIT_USER_TIME

---

**QUIES_WAIT_USER_TIME**

The QUIES_WAIT_USER_TIME statement enables you to define the time interval, in increments of 0.01 seconds, during a quiesce that RODM allows for a user to disconnect from RODM before RODM disconnects the user.

The syntax for the QUIES_WAIT_USER_TIME statement is:

```
QUIES_WAIT_USER_TIME
```

Where:

```
200
```

Specifies the time interval, in increments of 0.01 seconds, that RODM allows for a user to disconnect from RODM before RODM disconnects the user.

**Usage Notes:**

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.

**Related Statements:** QUIES_WAIT_USER_TIME
Where:

200|nnnn
Specifies the time interval, in increments of 0.01 seconds, that RODM allows for a user to disconnect during a quiesce before RODM disconnects the user. You can specify a value in the range of 1–32767. The default is 200.

Usage Notes:

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.

Related Statements: QUIES_WAIT_TASK_TIME

SEC_CLASS

The SEC_CLASS statement enables you to define the security class name used by your system authorization facility (SAF) product. Code this statement to correspond to your SAF product.

The syntax for the SEC_CLASS statement is:

```
SEC_CLASS
```

Where:

RODMMGR|classname
Specifies the security class name used by your SAF product. This name is restricted to a maximum length of 8 characters. The default is RODMMGR.

*TSTRODM
When specified for classname, indicates that RODM is not to issue SAF security checks. *TSTRODM is useful if you are testing a second copy of RODM and your SAF product is active, or do not require security for RODM.

Usage Notes:

- If you do not define the class name in EKGCUST or if the EKGCUST DD statement is not included in the JCL, the default security class name is used.
- Use the SEC_CLASS operand if you are using any security management system. For example, if you are using the Resource Access Control Facility (RACF) as your security management system, supply a SEC_CLASS name to RACF for use by RODM. If RACF is active, a check is made to see if a class is defined in EKGCUST. RACF authorization checking is handled as follows:
  - When RACF is not active, all the users have the highest authority level to RODM.
  - During RODM initialization, any one of these conditions causes RODM to stop processing with an error message:
If RACF is active, and the class you specified in EKGCUST is not active in RACF

If RACF is active, and the class you specified in EKGCUST is not defined in RACF

If RACF is active, you did not specify the class in EKGCUST, and the default class RODMMGR is not active in RACF

In these cases, RODM continues initialization with an informational message and you are not able to connect to RODM:

If RACF is active, the class you specified in EKGCUST is active in RACF, and the SEC_RNAME you specified in EKGCUST is not defined under the class in RACF

If RACF is active, you did not specify the class in EKGCUST, the default class RODMMGR is active in RACF, and the SEC_RNAME you specified in EKGCUST is not defined under the class in RACF

If RACF is active, the class you specified in EKGCUST is active in RACF, you did not specify the SEC_RNAME in EKGCUST, the default RODM name is used as SEC_RNAME, and this name is not defined under the class in RACF

If RACF is active, you did not specify the class in EKGCUST, the default class RODMMGR is active in RACF, you did not specify the SEC_RNAME in EKGCUST, the default RODM name is used as SEC_RNAME, and the RODM name is not defined under the RODMMGR class in RACF.

Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.

This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: SEC_RNAME

SEC_RNAME

The SEC_RNAME statement enables you to specify a prefix for the resource names that are used to check the six RODM authority levels, as shown in Table 17 on page 345 in the SAF product. The resource name will be the prefix you specify with a digit (1–6) appended. For example, if the SEC_RNAME statement was specified as SEC_RNAME(RODM), the resource names in the SAF product would be:

RODM1
RODM2
RODM3
RODM4
RODM5
RODM6

The syntax for the SEC_RNAME statement is:

SEC_RNAME

Where:
**SEC_RNAME**

Specifies the prefix for the resource name used by your SAF product. The valid length for the prefix is in the range of 1–43 characters.

**Usage Notes:**
- If you do not specify SEC_RNAME, but you specify a name for your RODM on your START command, the name that you specify is used to create the security resource names.
- If you do not specify SEC_RNAME, and you do not specify a name for your RODM on your START command, the name of the RODM start procedure is used to create your security resource names.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

**Table 17. RODM Access Authority Levels**

<table>
<thead>
<tr>
<th>Authority Level</th>
<th>Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connecting</td>
</tr>
<tr>
<td>2</td>
<td>Query and list of functions (queries only)</td>
</tr>
<tr>
<td>3</td>
<td>Action and list of functions (queries or actions) including triggering methods</td>
</tr>
<tr>
<td>4</td>
<td>Checkpointing</td>
</tr>
<tr>
<td>5</td>
<td>Administrative functions</td>
</tr>
<tr>
<td>6</td>
<td>Stopping</td>
</tr>
</tbody>
</table>

**Related Statements:** SEC_CLASS

**SEGMENT_POCKETS**

The SEGMENT_POCKETS statement enables you to define the number of backup translation segments allocated when RODM starts. When a segment is used up, RODM makes a request to get another segment. While the request is being processed, RODM uses the backup translation segment.

The syntax for the SEGMENT_POCKETS statement is:

```
SEGMENT_POCKETS (1nn)
```

**Where:**

\[1\] \(1\leq nn \leq 16\)

Specifies the number of backup segments. You can specify a value in the range of 1–16. The default is 1.

**Usage Notes:**
Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.

This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: MAX_WINDOW_NUM, WINDOW_POCKETS

SSB_CHAIN

The SSB_CHAIN statement enables you to define the number of same-name system status blocks (SSBs) that can concurrently exist in the system.

SSB_CHAIN

Where:

3

Specifies the number of same-name system status blocks that can exist concurrently in the system. You can specify a value in the range of 1–32767. The default is 3.

Usage Notes:

- RODM maintains a chain of RODM activation records in the common storage area (CSA) of an MVS system. When an application requests services from RODM, this information is used to determine whether RODM is active. This statement controls the number of retained entries. If, during a cold-start or a warm-start, RODM detects that the limit has been reached, RODM releases all inactive SSBs associated with the RODM name that is processing the SSB chain.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: QUIES_WAIT_TASK_TIME

TRANSPARENT_CHECKPOINT

The TRANSPARENT_CHECKPOINT statement enables you to specify whether to queue up the UAPI and MAPI requests while RODM is checkpointing.

The syntax for the TRANSPARENT_CHECKPOINT statement is:

TRANSPARENT_CHECKPOINT

Related Statements: SEGMENT_POCKETS
TRANSPARENT_CHECKPOINT

Where:

NO
Specifies that transparent checkpoint is not enabled. User and method API requests are rejected with a checkpoint-in-progress condition and must be retried. The default is NO.

YES
Specifies that transparent checkpoint is enabled. User and method requests are queued until the checkpoint completes. When checkpoint completes, the requests are run. Applications do not have to implement retry logic for checkpoint-in-progress conditions.

Usage Notes:

- This is a reloadable statement. If you reload RODM, RODM uses the keyword values contained in the customization file you specify with the MVS MODIFY command.
- Changes to this statement are used in a warm start. If you warm start RODM, RODM uses the keyword values you specified in a customization file.

WAIT_ALLOC_TIME

The WAIT_ALLOC_TIME statement enables you to define the waiting time, in increments of 0.01 seconds, for allocating window segments before RODM interprets the attempt as a failure.

The syntax for the WAIT_ALLOC_TIME statement is:

WAIT_ALLOC_TIME

Where:

100|nnn
Specifies the time to wait, in increments of 0.01 seconds, for allocating windows of segments. You can specify a value in the range of 1–100. The default is 100.

Usage Notes:

- RODM submits an asynchronous request to allocate another segment or window. If the response is not returned during the time specified in this keyword, RODM considers the request as failed.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.
WINDOW_CHKPT_TIME

The WINDOW_CHKPT_TIME statement enables you to define the wait time, in increments of 0.01 seconds, for taking a checkpoint of a window before RODM interprets the checkpoint as a failure.

The syntax for the WINDOW_CHKPT_TIME statement is:

\[
\text{WINDOW_CHKPT_TIME} \quad (\frac{\text{1500}}{\text{nnnn}})
\]

Where:

\[
\text{1500 \mid nnnn}
\]

Specifies the time to wait, in increments of 0.01 seconds, for taking a checkpoint of a window. You can specify a value in the range of 1–1500. The default is 1500.

Usage Notes:

- RODM submits an asynchronous request to take a checkpoint on another segment or window. If the response is not returned during the time specified in this keyword, RODM considers the request as failed.
- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- If you reload RODM, RODM uses the keyword values contained in the customization file you specify in the MVS MODIFY command.

Related Statements: WAIT_ALLOC_TIME

WINDOW_POCKETS

The WINDOW_POCKETS statement enables you to define the number of backup windows allocated when RODM starts. When a window is used up, RODM makes a request to get another window. While the request is being processed, RODM uses the backup translation window.

The syntax for the WINDOW_POCKETS statement is:

\[
\text{WINDOW_POCKETS} \quad (\frac{1}{\text{nn}})
\]

Where:

\[
1 \mid nn
\]

Specifies the number of backup windows. You can specify a value in the range of 1–16. The default is 1.
Usage Notes:

- Changes to this statement are used in a warm start. If you warm-start RODM, RODM uses the keyword values you have specified in a customization file.
- This is not a reloadable statement. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

Related Statements: MAX_WINDOW_NUM, SEGMENT_POCKETS
WINDOW_POCKETS
Chapter 7. Graphic Monitor Facility Host Subsystem

Statements

The Graphic Monitor Facility host subsystem (GMFHS) initialization statements are contained in DUIGINIT. These statements are system controlling constants that are read when GMFHS is initialized. You can use the LISTINIT command to display the GMFHS initialization statement values.

API

The API statement specifies the application programming interface to be traced.

The syntax for the API statement is:

```
API
```

Where:

- **NONE** | api
  - Indicates the application programming interface to trace. The api value can be NONE (the default), ALL, or one or more of the following:
    - IPC
    - PPI
    - RCM
    - RODM

Usage Notes:

- The GMFHS initialization statements cannot span multiple lines.
- Do not code parentheses for a single operand.
- If you code a keyword that is not valid, message DUI4076E is issued and a default of NONE is used for initialization of GMFHS.
- If you do not code the API statement, a default of NONE is used for initialization of GMFHS.
- If ALL appears with any keyword other than NONE, ALL is used as the initialization value.
- If NONE appears with any keyword other than ALL, NONE is used as the initialization value.
- If ALL and NONE both appear as the initialization value, the value listed first is used, and the message DUI4076E is issued.
Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.

Related Statements: LEVEL, PRINTPDU38, STORAGE, TASK, TRACE, TYPE

CHECKPOINT

The CHECKPOINT statement specifies when GMFHS requests RODM checkpoints.

The syntax for the CHECKPOINT statement is:

```
CHECKPOINT
```

Where:

NONE
No checkpoints are taken. This keyword, if specified, is mutually exclusive with any others. NONE is the default.

ALL
Checkpoints are taken at the following occurrences:
- At GMFHS startup time
- At GMFHS ending time
- After a CONFIG statement is processed by GMFHS

This keyword, if specified, is mutually exclusive with any others.

STARTUP
A checkpoint is taken at GMFHS startup time.

TERM
A checkpoint is taken when GMFHS ends.

CONFIG
A checkpoint is taken after a CONFIG command is processed by GMFHS.

Usage Notes:
- Do not code parentheses for a single operand.
- If you do not code the CHECKPOINT statement, a default of NONE is used for initialization of GMFHS.
- If you code a keyword of STARTUP, a checkpoint is generated when a CONFIG NETWORK command is issued.
- If you code a keyword that is not valid, message DUI4022 is issued, and a default of NONE is used for initialization of GMFHS.
• If ALL or NONE appears with any other keywords, a conflict of keywords exists. Message DUI4022 is issued, and a default of NONE is used for initialization of GMFHS.

**DOMAIN**

The DOMAIN statement identifies the domain to which this GMFHS component belongs. This is a required statement.

The syntax for the statement is:

```plaintext
DOMAIN
```

Where:

- `domainid` Indicates the ID of the GMFHS component (a maximum of 5 alphanumeric characters).

Usage Note: DOMAIN is used to provide a unique identification of the GMFHS component. It should be set to the same value specified in the NetView address space. Note that GMFHS will not start if this parameter is not specified.

**GMTOFFSET**

The GMTOFFSET statement specifies the Greenwich Mean Time offset, which is the number of hours and minutes that the time differs from Greenwich Mean Time (also called Coordinated Universal Time, or UTC).

The syntax for the GMTOFFSET statement is:

```plaintext
GMTOFFSET
```

Where:

- `s` Indicates a positive (+) or negative (−) sign. A positive sign indicates that the offset must be added to the local time to produce the Greenwich Mean Time. A negative (−) sign indicates that the offset must be subtracted from the local time to produce the Greenwich Mean Time.

- `hh` Indicates the hours of the Greenwich Mean Time offset.

- `mm` Indicates the minutes of the Greenwich Mean Time offset.

Usage Notes:

- If you specify an incorrect value for the `hh` and `mm` values, message DUI4029 is issued and GMFHS ends.
GMTOFFSET

- The valid range is from −2359 to +2359. If you do not specify GMTOFFSET, the host MVS GMT time is used. Use the LISTINIT command to view the host MVS GMT time.

JAPANESE

The JAPANESE statement specifies whether or not GMFHS should use Japanese text for any displayed text that it provides to an NMC console.

The syntax for the JAPANESE statement is:

```
JAPANESE
```

Where:

**OFF**
- Disables translation to Japanese. This is the default.

**ON**
- Enables translation to Japanese.

LCON-AGG-BUNDLE-INTERVAL

The LCON-AGG-BUNDLE-INTERVAL statement specifies the time interval between invocations of DUIFCASB.

The syntax for the LCON-AGG-BUNDLE-INTERVAL statement is:

```
LCON-AGG-BUNDLE-INTERVAL
```

Where:

```
500
```

This indicates a value, in hundredths of a second, in the range from 10 to 864000000000. The default value specified in the sample is 500.

Usage Note: Because this is the only routine that changes Aggregate object counts, there is no need to explicitly lock lists of objects to ensure that asynchronous changes are made to an Aggregate currently being processed.
The LCON-AGGRST-REQUIRED statement enables you to specify whether you want the aggregation method to run at startup. A related parameter included on the GMFHS start procedure is AGGRST. If you specify AGGRST in the start procedure, its value overrides the value specified by the LCON-AGGRST-REQUIRED statement at GMFHS initialization. Thereafter, the LCON-AGGRST-REQUIRED value is used.

The syntax for the LCON-AGGRST-REQUIRED statement is:

```
LCON-AGGRST-REQUIRED
```

Where:

- **NO**
  Specifies not to run the method. This is the default.
- **YES**
  Specifies to run the DUIFFAWS aggregation warm-start method at startup and after a CONFIG NETWORK command.

The LCON-AIP-RESET-INTERVAL statement specifies the time interval between invocations of an object independent method, which is triggered to obtain a list of real resource objects and has the AIP bit set in the UserStatus field.

The syntax for the LCON-AIP-RESET-INTERVAL statement is:

```
LCON-AIP-RESET-INTERVAL
```

Where:

- **12000 | interval**
  Is a value, in hundredths of a second, in the range from 0 to 864 000 000 000. The minimum value of zero (0) indicates that GMFHS will not monitor the AIP bit to determine if it should be reset. The maximum value of 864000000 is equivalent to 100 days. The default value specified in the sample is 12000, which is equivalent to 2 minutes.

Usage Notes:

- This method obtains a list of the real resource objects with the AIPTIMER bit set. The AIPTIMER bit indicates that this resource had the AIP bit set the last time the method was triggered.
Resources need to have the AIPTIMER bit set on each time the method is monitored to ensure that the AIP bit does not hang. The interval specifies the time between invocations of method DUIFRAIP which determines if the AIP bit should be reset.

**LCON-ALERT-CMD-TIMEOUT**

The LCON-ALERT-CMD-TIMEOUT statement specifies the number of seconds, multiplied by 0.01, Dbservice waits before terminating the alert history request of an NetView management console client waiting for reply data. The time-out while waiting for data that may occur at the NetView management console client is unaffected by this setting.

The syntax for the LCON-ALERT-CMD-TIMEOUT statement is:

```
LCON-ALERT-CMD-TIMEOUT
```

Where:

\[
\text{timeout} = 30000 \times 0.01
\]

Indicates a value, in hundredths of a second, in the range from 100–360000. The default value specified in the sample is 30000.

**LCON-ASSOCIATE-NULL-NODE-WITH-LINK**

The LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement specifies how null nodes will be built. All links must have two end points. If an endpoint does not exist, a null node is created and used as an endpoint for the link. Each null node has a unique RODM object id.

The syntax for the LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement is:

```
LCON-ASSOCIATE-NULL-NODE-WITH-LINK
```

Where:

- **0** Default. The object id of a null node does not specify any relationship to the link for which it was created. This may result in potential layout problems for null nodes in customized views. However, changing the default value may result in problems migrating customized views from an earlier release.

- **1** The object id of a null node associates that node with the correct link, thus avoiding potential layout problems for null nodes in customized views. The only downfall with this approach is migration; any existing customized views that contain null nodes will be lost.

Usage Note:
LCON-EVCHANGE-BUFFER-INTERVAL

Changing the default value may result in problems migrating customized views.

LCON-EVCHANGE-BUFFER-INTERVAL

The LCON-EVCHANGE-BUFFER-INTERVAL statement specifies the time interval between invocations of the resource traits change method.

The syntax for the LCON-EVCHANGE-BUFFER-INTERVAL statement is:

LCON-EVCHANGE-BUFFER-INTERVAL

Where:

\[ \text{interval} \]

Indicates a value, in hundredths of a second, in the range from 10 to 864 000 000 000. The default value specified in the sample is 500.

Usage Note: When the interval has expired, the resource traits change method will be triggered to bundle view updates. The view updates consist of data relevant to Display or User status changes or ExceptionViewFilter or ExceptionViewList fields.

Related Statements: None.

LCON-HEX-SUBVECTOR-DISPLAY

The LCON-HEX-SUBVECTOR-DISPLAY statement controls the display of the hexadecimal dump of the subvectors in a major vector as part of the response to the alert explanation command.

The syntax for the LCON-HEX-SUBVECTOR-DISPLAY statement is:

LCON-HEX-SUBVECTOR-DISPLAY

Where:

\[ \begin{align*}
01 & \quad \text{Indicates that a display of the hexadecimal dump of subvectors is to be generated. You can also specify this value as 1. The default value is 01.} \\
00 & \quad \text{Indicates that a display of the hexadecimal dump of subvectors is not to be generated. You can also specify this value as 0.}
\end{align*} \]

Usage Notes:
• If you do not code this statement in DUIGINIT, the default of 01 is used.
• If you specify a number other than 00 or 01, message DUI4074E is issued, and the default of 01 is used.
LCON-MAX-LOCATE-RESOURCE

**LCON-MAX-LOCATE-RESOURCE-VIEWS**

The LCON-MAX-LOCATE-RESOURCE-VIEWS statement specifies the maximum number of views that will be returned on a Locate Resource request.

The syntax for the LCON-MAX-LOCATE-RESOURCE-VIEWS statement is:

```
LCON-MAX-LOCATE-RESOURCE-VIEWS
```

Where:

```
12 | number_of_views
```

- Specifies the maximum number of views that will be returned on a Locate Resource request. Valid values are in the range from 1 to 32767. The default value is 12.

Usage Note:

Increasing the `number_of_views` value may cause timeouts on the workstation.

**LCON-MAX-QUEUE-DBSERVER**

The LCON-MAX-QUEUE-DBSERVER statement specifies the maximum number of messages that can be held on the Dbserver task message queue.

The syntax for the LCON-MAX-QUEUE-DBSERVER statement is:

```
LCON-MAX-QUEUE-DBSERVER
```

Where:

```
10000 | queue_size
```

- Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:

- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- You might have to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.

Related Statements: LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC, LCON-MAX-QUEUE-IRMGR,
LCON-MAX-QUEUE-EVENTMGR

The LCON-MAX-QUEUE-EVENTMGR statement specifies the maximum number of messages that can be held on the EVENTMGR task message queue.

The syntax for the LCON-MAX-QUEUE-EVENTMGR statement is:

```
LCON-MAX-QUEUE-EVENTMGR
```

Where:

```
queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- You might have to customize this statement if there is a possibility of sustained high alert traffic because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


LCON-MAX-QUEUE-IPC

The LCON-MAX-QUEUE-IPC statement specifies the maximum number of messages that can be held on the IPC task message queue.

The syntax for the LCON-MAX-QUEUE-IPC statement is:

```
LCON-MAX-QUEUE-IPC
```

Where:

```
queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- You might have to customize this statement if there is a possibility of sustained high alert traffic because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.

LCON-MAX-QUEUE-IPC

\[ \text{queue_size} \]

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:

- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- You might have to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


LCON-MAX-QUEUE-IRMGR

The LCON-MAX-QUEUE-IRMGR statement specifies the maximum number of messages that can be held on the IRMGR task message queue.

The syntax for the LCON-MAX-QUEUE-IRMGR statement is:

```
LCON-MAX-QUEUE-IRMGR
```

Where:

\[ \text{queue_size} \]

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:

- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- It might be necessary to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.

The LCON-MAX-QUEUE-MAINTASK statement specifies the maximum number of messages that can be held on the MAINTASK task message queue.

The syntax for the LCON-MAX-QUEUE-MAINTASK statement is:

```
LCON-MAX-QUEUE-MAINTASK
```

```
LCON-MAX-QUEUE-MAINTASK= queue_size
```

Where:

```
10000|queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

**Usage Notes:**
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- You might have to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


The LCON-MAX-QUEUE-NETCMD statement specifies the maximum number of messages that can be held on the NETCMD task message queue.

The syntax for the LCON-MAX-QUEUE-NETCMD statement is:

```
LCON-MAX-QUEUE-NETCMD
```

```
LCON-MAX-QUEUE-NETCMD= queue_size
```

Where:

```
10000|queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.
**LCON-MAX-QUEUE-NETCMD**

*Usage Notes:*
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 1000 is used.
- It might be necessary to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


**LCON-MAX-QUEUE-NETCON**

The LCON-MAX-QUEUE-NETCON statement specifies the maximum number of messages that can be held on the NETCON task message queue.

The syntax for the LCON-MAX-QUEUE-NETCON statement is:

```
LCON-MAX-QUEUE-NETCON
```

Where:

```
10000|queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

*Usage Notes:*
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- It might be necessary to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.

LCON-MAX-QUEUE-OPERIF

The LCON-MAX-QUEUE-OPERIF statement specifies the maximum number of messages that can be held on the OPERIF task message queue.

The syntax for the LCON-MAX-QUEUE-OPERIF statement is:

```
LCON-MAX-QUEUE-OPERIF
```

Where:

```
10000|queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:

- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- It might be necessary to customize this statement if there is a possibility of your disk constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


LCON-MAX-QUEUE-RCMGR

The LCON-MAX-QUEUE-RCMGR statement specifies the maximum number of messages that can be held on the RCMGR task message queue.

The syntax for the LCON-MAX-QUEUE-RCMGR statement is:

```
LCON-MAX-QUEUE-RCMGR
```

Where:

```
10000|queue_size
```

Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.
LCON-MAX-QUEUE-RCMGR

Usage Notes:

- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 1000 is used.
- It might be necessary to customize this statement if there is a possibility of your disk constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


LCON-MAX-QUEUE-RTMGR

The LCON-MAX-QUEUE-RTMGR statement specifies the maximum queue size for the Resource Trait subtask.

The syntax for the LCON-MAX-QUEUE-RTMGR statement is:

```
LCON-MAX-QUEUE-RTMGR
```

Where:

```
10000
```

Indicates a value in the range from 100 to 10000. The default value specified in the sample is 10000.

Usage Note: If you specify a value greater than 10000, message DUI4074E is issued and the default of 1000 is used.


LCON-MAX-QUEUE-VIEWMGR

The LCON-MAX-QUEUE-VIEWMGR statement specifies the maximum number of messages that can be held on the VIEWMGR task message queue.

The syntax for the LCON-MAX-QUEUE-VIEWMGR statement is:

```
LCON-MAX-QUEUE-VIEWMGR
```
LCON-MAX-QUEUE-VIEWMGR

Where:

10000 | queue_size
Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- It might be necessary to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.


LCON-MAX-QUEUE-VSTATMGR

The LCON-MAX-QUEUE-VSTATMGR statement specifies the maximum number of messages that can be held on the VSTATMGR task message queue.

The syntax for the LCON-MAX-QUEUE-VSTATMGR statement is:

LCON-MAX-QUEUE-VSTATMGR

Where:

10000 | queue_size
Indicates a value in the range from 100 to 65535. The default value specified in the sample is 10000.

Usage Notes:
- If you specify a value outside the range for this statement, message DUI4074E is issued and the default of 10000 is used.
- It might be necessary to customize this statement if there is a possibility of your disks constantly being accessed because of asynchronous events. If this activity is sustained for long periods, queued messages can increase. Results are unpredictable if message queues are allowed to increase without restraint.
LCON-NCC-RETRY-LIMIT

The LCON-NCC-RETRY-LIMIT statement specifies the number of times GMFHS attempts to resend a status solicitation network command to a native element manager.

Note: LCON-NCC-RETRY-LIMIT does not apply to commands issued from the NetView management console.

The syntax for the LCON-NCC-RETRY-LIMIT statement is:

```plaintext
LCON-NCC-RETRY-LIMIT
```

Where:

- `3 | retry_limit`
  - Indicates a value in the range from 0 to 99. If you specify 0, status solicitation retries are not attempted. The default value is 3.

Usage Notes:

- If the manager of the transport facilities involved indicates that a command has timed out or could not be processed because of a temporary condition (for example, Resource Busy or Currently Not Available), LCON-NCC-RETRY-LIMIT enables you to specify the number of times to attempt to resend the command. If this limit is exceeded, GMFHS ends the status solicitation for the element manager.
- If you use a value less than zero (0), message DUI4074E is issued and the default of 3 is used at GMFHS initialization. If you use a value greater than 99, messages DUI4074E and DUI4075E are issued and the default of 3 is used at GMFHS initialization.

Related Statements: LCON-NCC-RSC-LIMIT

LCON-NCC-RSC-LIMIT

The LCON-NCC-RSC-LIMIT statement specifies the maximum number of resources that the network command manager task can name in a status solicitation network command (for example, Display Status) for initial status for those native element managers that use the DOMP010 protocol. This group of resources can be the set, a subset, or all of the resources for a particular native element manager. After status has been solicited for this subset, the status for the next subset of resources is solicited until all resources for the native element manager have had their status solicited.

Related Statements: LCON-NCC-RSC-LIMIT
Note: LCON-NCC-RCS-LIMIT does not apply to commands issued from the NetView management console workstation.

The syntax for the LCON-NCC-RSC-LIMIT statement is:

```
LCON-NCC-RSC-LIMIT
```

Where:

```
10 | resource_limit
```

Indicates a value in the range from 1 to 99. The default value is 10.

Usage Notes:

- The maximum resource_limit must be a value that does not cause a message to be generated that exceeds the transport or gateway capacity. The size of the message varies with the size of the native network resource names used in communicating with the element manager.
- If you specify a negative number, message DUI4074E is issued, and the default of 10 is used. If you specify 0 or a number greater than 99, messages DUI4074E and DUI4075E are issued, and the default of 10 is used.

Related Statements: LCON-NCC-RETRY-LIMIT

---

LCON-NMG-POLL-INTERVAL

The LCON-NMG-POLL-INTERVAL statement specifies the interval, in hundredths of a second, between command session establishment polls for those native element managers that use the DOMS010 session protocol, and for which a command session has not been established.

The syntax for the LCON-NMG-POLL-INTERVAL statement is:

```
LCON-NMG-POLL-INTERVAL
```

Where:

```
18000 | interval
```

Indicates a value, in hundredths of a second, in the range from 100 to 360000. The default value is 18000 (3 minutes).

Usage Notes:

- This value should allow sufficient, but not excessive, time for a session request message to be generated and delivered to an element manager, and for the response to return to the GMFHS. If you set this value for too short an interval, a new session request might be sent to the element manager before GMFHS
receives the INIT alert that was issued in response to the first request. Multiple
INIT alerts received from an element manager can cause status solicitation to fail
and restart repeatedly.

* If you specify a number outside the valid range, messages DUI4074E and
DUI4075E are issued, and the default of 18000 is used.

**LCON-OPERATOR-CMD-AUDIT**

The LCON-OPERATOR-CMD-AUDIT statement specifies that operator commands
received from a workstation and responses sent to the workstation are written to
an audit log. The audit log contains the ID of the operator that issued the
command, the domain that will receive the command, the route LU name (if
applicable), a time stamp of when the command was sent, and the command text.
The audit log also contains the response text, if a response is expected and
received.

The syntax for the LCON-OPERATOR-CMD-AUDIT statement is:

```
LCON-OPERATOR-CMD-AUDIT
```

Where:

- **00**: Indicates that command audit tracing is off. You can also specify this value as
  zero (0). This is the default.
- **01**: Indicates that command audit tracing is on. You can also specify this value as
  1.

**Usage Note:** If you specify a number other than 0, 00, 1, or 01, messages DUI4074E
and DUI4075E are issued, and the default of 00 is used at GMFHS initialization.

**LCON-REPORT-UNKNOWN-STATUS**

The LCON-REPORT-UNKNOWN-STATUS statement controls logging of the
system error synopsis protocol data units (PDUs) when GMFHS cannot translate
an alert status. This condition occurs when the alert type is not identified in either
of the alert type translation tables (DUIFEIBM and DUIFEUSR) supplied with
GMFHS.

The syntax for the LCON-REPORT-UNKNOWN-STATUS statement is:

```
LCON-REPORT-UNKNOWN-STATUS
```

Where:
### LCON-REPORT-UNKNOWN-STATUS

**00** Indicates that system error synopsis PDUs are not logged. You can also specify this value as 0. This is the default.

**01** Indicates that system error synopsis PDUs are logged and MVS console message DUI3913 is issued. You can also specify this value as 1.

**Usage Note:** If you specify a number other than 0, 00, 1, or 01, message DUI4074E is issued and the default of 00 is used at GMFHS initialization.

### LCON-SNATM-TIMEOUT

The LCON-SNATM-TIMEOUT statement indicates how long (in hundredths of seconds) the Graphic Monitor Facility Host Subsystem (GMFHS) will wait before timing out, for the SNA Topology Manager response to a locate resource request.

The syntax for the LCON-SNATM-TIMEOUT statement is:

```
LCON-SNATM-TIMEOUT
```

**Where:**

\[12000 \text{ interval}\]

Indicates a value, in hundredths of a second, in the range from 100 to 360000. The default value is 12000.

**Usage Notes:**

- If you increase the LCON-SNATM-TIMEOUT value you might also need to increase the communication time-out values on the NetView management console workstations.
- If you specify a number outside the valid range, message DUI4074E is issued, and the default of 12000 is used.

### LCON-STATUS-DELAY-MAX

The LCON-STATUS-DELAY-MAX statement specifies the maximum number of LCON-STATUS-DELAY-TIME intervals that GMFHS uses before it sends status updates to the NetView management console.

The LCON-STATUS-DELAY-MAX statement also specifies the maximum number of LCON-STATUS-DELAY-TIME intervals that GMFHS uses before it notifies workstations of changes to specific views.

The syntax for the LCON-STATUS-DELAY-MAX statement is:

```
LCON-STATUS-DELAY-MAX
```

**Where:**

\[10 \text{ count}\]

Indicates a number, in the range from 1 to 10. The default value is 10.
LCON-STATUS-DELAY-MAX

Where:

10 | count

Specifies the maximum number of LCON-STATUS-DELAY-TIME intervals. This is a value in the range from 0 to 50. The default value is 10.

Usage Note: If you specify a value greater than 50, messages DUI4074E and DUI4075E are issued, and the default of 10 is used at GMFHS initialization. If you specify a value less than zero (0), message DUI4074E is issued and the default of 10 is used at GMFHS initialization.

Related Statements: LCON-STATUS-DELAY-TIME

LCON-STATUS-DELAY-TIME

The LCON-STATUS-DELAY-TIME statement specifies the maximum time, in hundredths of a second, during which GMFHS accumulates status updates for views opened in the NetView management console.

The LCON-STATUS-DELAY-TIME statement also specifies the maximum time interval that GMFHS uses before it notifies workstations of changes to specific views. The time interval that GMFHS waits before sending a list of the changed views is approximately 10 times greater than the value specified in the LCON-STATUS-DELAY-TIME statement.

The syntax for the LCON-STATUS-DELAY-TIME statement is:

LCON-STATUS-DELAY-TIME

Where:

50 | interval

Specifies the maximum time, in hundredths of a second, during which GMFHS accumulates status updates for views opened in the graphic monitor. This is a value in the range from 10 to 864000. The default value is 50 (0.5 seconds).

Usage Notes:

- If new status updates are received in this interval, GMFHS accumulates and waits additional intervals up to the LCON-STATUS-DELAY-MAX interval before sending the accumulated status updates to the graphic monitor.
- If you specify a value outside the valid range, messages DUI4074E and DUI4075E are issued, and the default of 50 is used at GMFHS initialization.

Related Statements: LCON-STATUS-DELAY-MAX

LEVEL

The LEVEL statement indicates the minimum level of program trace statement that is included in each trace entry.

The syntax for the LEVEL statement is:
LEVEL

Where:

\(0 \leq \text{lvl} \leq 99\)

Specifies the minimum level of program trace statement that is included. The range is 0 to 99. The default value for the LEVEL statement is zero (0).

Usage Notes:

- The LEVEL statement value is interpreted by each task listed in the TASK statement and is used to control the amount of detail in each trace entry.
- If you specify a value that is out of range, message DUI4076E is issued, GMFHS initialization continues, and the default value of zero (0) is used.
- You cannot use parentheses. If they are present, message DUI4008E is issued, and the default of zero (0) is used.
- Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.

Related Statements: API, PRINTPDU38, STORAGE, TASK, TRACE, TYPE

PRINTPDU38

The PRINTPDU38 statement specifies where to log system error synopsis (PDU38) information.

The syntax for the PRINTPDU38 statement is:

```
PRINTPDU38
```

Where:

**YES or FILE**

Issued to the GMFHS output data sets. The output data sets are defined by GMFHS using the following DD statement in the GMFHS startup procedure:

- CNMC  Network command manager (NETCMD)
- CNMD  Database server (DBSERVER)
- CNME  Event manager (EVENTMGR)
- CNMF  Network configuration manager (NETCON)
- CNMI  Interprocessor communication (IPC)
- CNMM  GMFHS main (control) task (MAINTASK)
- CNMO  Operator interface manager (OPERIF)
**PRINTPDU38**

<table>
<thead>
<tr>
<th>CNMP</th>
<th>IPC-RODM event manager (IRMGR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNMR</td>
<td>Resource traits manager (RTMGR)</td>
</tr>
<tr>
<td>CNMS</td>
<td>View status manager (VSTATMGR)</td>
</tr>
<tr>
<td>CNMV</td>
<td>View manager (VIEWMGR)</td>
</tr>
</tbody>
</table>

**NO or INTERNAL**

Issued to the GMFHS internal trace log.

**GTF**

Issued to GTF.

The event identifier (EID) used for the TRACE records written to GTF is X'5E2'.

**Usage Notes:**

- If you specify a value that is not valid, message DUI4076E is issued, and the default of NO or INTERNAL is used at GMFHS initialization.
- Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.
- PDU38 logging cannot be deactivated. PDU38 information is written to the GMFHS internal trace log unless otherwise specified on the PRINTPDU38 statement.
- If the internal trace log is deactivated and none of the other output logs are available to log PDU38 data, the internal trace log will be temporarily activated. As soon as one of the other output logs become available, the internal trace log will be deactivated.

**Related Statements:** API, LEVEL, STORAGE, TASK, TRACE, TRACEBYTES, TRACEPAGES, TYPE

---

**RODMID**

The RODMID statement identifies the RODM user ID to be used in the attempt to connect to the RODM you specified in RODMNAME. This is a required statement.

The syntax for the RODMID statement is:

```
RODMID
```

Where:

rodmid

Indicates the ID of the RODM user (a maximum of 8 characters).

**Usage Notes:**

RODMID is used to provide a unique identification of the autotask as a RODM user.
- If you have an SAF product (such as RACF) installed, RODM uses the values provided by RODMID to determine if the autotask is authorized to connect to the RODM specified by the RODMNAME statement.
If you do not have an SAF product (such as RACF) installed, the values provided by the RODMID statements are used only to provide unique identification for the autotask. In this case the autotask has unrestricted access to RODM services.

**Related Statements:** RODMNAME

**RODMNAME**

The RODMNAME statement specifies the name of the RODM that GMFHS uses. This is a required statement.

The syntax for the RODMNAME statement is:

```
RODMNAME
```

**Where:**

**rodmname**

Indicates the name of the RODM (a maximum of 8 characters).

**Usage Note:** If you have a security access facility product (such as RACF) installed, RODM uses the values provided by RODMID to determine if the autotask is authorized to connect to the RODM specified by the RODMNAME statement.

**Related Statements:** RODMID

**STORAGE**

The STORAGE statement specifies whether get and free storage tracing is on or off.

The syntax for the STORAGE statement is:

```
STORAGE
```

**Where:**

**NO**

Disables tracing for storage requests. This is the default.

**YES**

Enables tracing for storage requests.

**Usage Notes:**

- If you specify a value other than YES or NO, message DUI4076E is issued, and the default value of NO is used at GMFHS initialization.
- Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any
RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.

**Related Statements:** API, LEVEL, PRINTPDU38, TASK, TRACE, TYPE

### TASK

The TASK statement indicates the NetView GMFHS tasks to be traced.

The syntax for the TASK statement is:

```
TASK
```

Where:

- **task**: Indicates the GMFHS task to be traced. The task can be one or more of these tasks:
  - ALL
  - DBSERVER
  - EVENTMGR
  - IPC
  - IRMGR
  - MAINTASK
  - NETCMD
  - NETCON
  - NONE
  - OPERIF
  - RCMGR
  - RTMGR
  - VIEWMGR
  - VSTATMGR

**Usage Notes:**
- If you specify a value that is not valid, message DUI4076E is issued, and the value of NONE is used at GMFHS initialization.
- Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.

**Related Statements:** API, LEVEL, PRINTPDU38, STORAGE, TRACE, TYPE

### TRACE

The TRACE statement turns on the NetView GMFHS trace options. ON is the only correct choice if the GMFHS trace is not active. The TRACE statement is initially set to TRACE=OFF.

**Related Statements:** API, LEVEL, PRINTPDU38, STORAGE, TRACE, TYPE
Attention: Logging trace information to the VSAM database can adversely effect performance in logging and in accessing alert information.

The syntax for the TRACE statement is:

```
TRACE
```

Where:

**OFF**
Disables tracing for all GMFHS tasks specified with the TASK statement. This is the default.

**ON**
Enables tracing for all GMFHS tasks specified with the TASK statement.

Usage Notes:
- The first TRACE command you issue does not activate or deactivate the tracing facility; it only determines whether the tasks specified using the TASK statement are to be traced. The second TRACE command you issue activates or deactivates tracing. Refer to the NetView online help for more information about the TRACE command.
- If you specify a value other than YES or NO, message DUI4076E is issued, and the default value of OFF is used at GMFHS initialization.
- Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume, if possible.

Related Statements: API, LEVEL, PRINTPDU38, STORAGE, TASK, TYPE

### TRACEBYTES

The TRACEBYTES statement enables you to specify how many bytes of each trace record are written to the internal trace log. This is an optional statement.

The syntax for the TRACEBYTES statement follows:

```
TRACEBYTES
```

Where:

```
0 | bytes
```
- Indicates the number of bytes from each trace record that is written to the
TRACEBYTES

internal trace log. You can specify a value in the range from 32 to 32656. Zero (0) is also a valid value. The default value is 0.

Usage Note: It is recommended that you specify a number of TRACEBYTES that fits evenly into a 4 KB page.

Related Statements: TRACE, TRACEPAGES

TRACEPAGES

The TRACEPAGES statement enables you to specify the number of pages of storage to allocate for the internal trace log for GMFHS. This is an optional statement.

The syntax for the TRACEPAGES statement is:

```
TRACEPAGES
```

Where:

```
pages
```

Indicates the number of 4 KB pages to allocate for the internal trace log. The range of values for `pages` is 100 to 999. Zero (0) is also a valid value.

Usage Notes:

- The number of pages to allocate depends on your system. To prevent the loss of data when issuing a GMFHS TRACE FLUSH command, a new internal trace log of `pages` size is allocated before GMFHS prints and releases the current table.
- When determining the number of pages to allocate, remember that if you are tracing at higher levels, trace records are written to the trace table faster and cause the trace entries to wrap sooner.

Related Statements: LEVEL PRINTPDU38, TRACE, TRACEBYTES

TYPE

The TYPE statement instructs the tasks to include trace entries for specific interface and message types.

The syntax for the TYPE statement is:

```
TYPE
```

Where:

```
type
```

Indicates one of these interface and message types:
• ALL
• CNMTAMEL
• GDS
• NONE
• NOTIFY
• PDU
• PPI
• SCO

Usage Notes:
• If you specify a value that is not valid, message DUI4076E is issued, and the default value of NONE is used at GMFHS initialization.
• Tracing for GMFHS generates a large amount of output. Under high volume or stress conditions, tracing can severely degrade response in GMFHS and in any RODM methods used by GMFHS. Do not invoke tracing unless necessary for debugging, and invoke tracing under low volume.

Related Statements: API, LEVEL, PRINTPDU38, STORAGE, TASK, TRACE
Index

Special Characters
%INCLUDE statement 59
*PRE keyword, KCLASS statement 67
*SEC keyword, KCLASS statement 67

Numerics
3270 management console 78

A
A (Alert) statement 9
ACBNAME keyword, VPDINIT statement 205
access control block (ACB) primary data set 43
secondary data set 44
ACCESS statement 11
accessibility information xiii
action area, command facility panel 55
action messages, command facility panel 12
ACTION statement 12
ACTMON 209
AdapterCdsFile statement 305
AdapterFmtFile statement 307
additional status information, obtaining 182
addresses in the keyword list, F (Filter) statement 48, 49
ADJNETV 214
administration, system 9
AFTER command 27
Aggregate object counts, changing 354
AIPTIMER bit 355
AL keyword, W (Wrap) statement 137
alarm operands 59
ALCACHE examples, table 120, 121
statement 119
alert
cache size 119
focal point 37, 124
forwarding protocol 121
log 123
NPDA.ALERTLOG statement 123
record 124
alert receiver 172
ALERTFWD statement 121
ALERTLOG statement 123
AlertRcvName statement 13
alias name assignment, logical unit 73
translation tables 13
ALIASMEM statement 13
ALRITCFG statement 307
ALRITINFP statement 124
ALRITCFG statement 308
ALT_ALERT statement 125
AMELINIT statement 14
amplitude hits 128, 129
AMS 45
AON automation operator names 215
AON Control File Entries, Defining 209
ACTMON 209
ADJNETV 214
AUTOOPS 214
CDLOG 219
CPCPSES (SNA) 221
DDGENERIC 222
DDGROUP 223
ENVIRON AIP 224
ENVIRON CONSOLE 225
ENVIRON DDF 226
ENVIRON EXIT 228
ENVIRON RACF 229
ENVIRON SETUP 231
ENVIRON SNBU (SNA) 234
ENVIRON TIMEOUT 235
EZTLLOG 236
FORWARD FOCALPT 238
FULLSESS 239
GATEWAY 239
INSTALLOPT 242
IPCONN 243
IFHOST 245
IPINFC 246
IPNAMESERV 248
IPPORT 250
IPROUTE 253
IPTN3270 255
LSTTHRESH 257
MONIT 258
MONITOR 259
NCPRECOV 260
NOTIFY 263
NTFYOP 264
NV6000 (TCP/IP) 267
RECOVERY 269
RESIDENT control file entries in the control file 274
SESSION (SNA) statement for sessions monitored by AON/SNA 274
SNBU (SNA) 275
SNBUPool (SNA) 278
SUBSYSTEM control file entry for NVAS subsystem interface with SNA
help desk 279
TCP/IP network resources to AON/TCP 282
TCP/IP control file entry 279
TGSWITCH (SNA) 283
THRESHOLDS, setting 284
TIMER control file entries 287
TSERV 291
X25MONIT 291
AON Message Classes 265
API statement 351
APPCCMD 165
application name
CNM data transfer task 74
NetView 87
ASSIGN command 16
ASSIGN statement 14
ASYNC_TASKS statement 329
AT command 27
ATCCONxx
STATOPT statement 185
AUTH statement
usage 15
authorization checking commands and command lists 25
authorize remote operators 167
authorized receiver 16, 47
AUTOCCMD statement 20
AUTOFLIP keyword, LOGINIT statement 72
automatic re-activation 147
automating MSUs 126
automation operator names 215
automation table 20
automation task operator 150
AUTOOPS 214
AUTOFLIP statement 126
autotask, define to RODM access and control task 191
AUTOTASK command 150
autotask statement 50
AUTOTASK statement 21
autotasks 21
auxinitCmd statement 22
B
BACKUP keyword, DEFFOCPT statement 38
banner statement 22
BER data 131
BINDFAIL keyword, KCLASS statement 66
BLOCK keyword
RMFSEC statement 168
block remote operators 167
BNJMBS sample DBFULL statement 35
books
feedback xii
online xii
ordering xii
BOUNDS keyword, PCLASS statement 156
BufEvtMaxSize statement 309
BufEvtPath statement 309
BufEvtRlcklen statement 310
BufEvtShrinkSize statement 310
buffer size definition for reading data from event buffer 310
BufferEvents statement 311
C (Command List) statement 23
CAF interfaces 34
CCDEF statement 23
CDLOG 219
CDRMDEF statement 86
CDRSC, VTAM definition statement 73
CELL_POOLS statement 329
character translation sets 202, 203
CHARACTER_VALIDATION statement 331
CHECKPOINT_FUNCTION statement 332
CHECKPOINT statement 352
class keep 64, 76 of service 34 performance 76
Classes, AON Messages 265
CLOSE command 45
CMD keyword, RMTSEC statement 168
class CHECKPOINT statement 352
AUTOCMD statement 20
AUTOTASK statement 21
auxinItCmd statement 22
banner statement 22
CCDEF statement 23
COMMON statement 32
DEFAULTS statement 35
DOMAIN statement 40
function.autotask statement 50
GBH.TCPNAME statement 51
HARDCOPY statement 51
HLL statement 53
idleParms statement 57
INIT.NRM statement 61
inStore statement 62
IPOG statement 63
JesJobLog 64
LOGONFW statement 73
LUC.CNMARTG statement 74
LUC.CTL statement 75
LUC.MAXESS statement 75
LUC.PERSIST statement 75
LUC statement 74
MCON statement 78
memStore statement 79
MVS.PARM.DEAUTH 81
MVS.PARM.MIGRATE 82
MVS.PARM.MSIGFAC 83
NetID statement 84
NLD.M.AMLUTDLY statement 84
NLD.M.AUTHDOM statement 85
NLD.M.AUTHORIZ statement 86
NLD.M.CDRMDEF statement 86
NLD.M.CDTIME statement 87
NLD.M.DRDELAY statement 88
NLD.M.DSRBO statement 88
NLD.M.ERCOUNT statement 89
NLD.M.FCTIME statement 89
NLD.M.KEEPDISC statement 90
NLD.M.KEEPFCLASS statement 90
NLD.M.KEEPFCLASS statement 90
NLD.M.KEEPFPLU statement 90
NLD.M.KEEPFRTM statement 91
NLD.M.KEEPSESS statement 91
NLD.M.LOG statement 92
NLD.M.LUCOUNT statement 93
NLD.M.MACRF statement 93
NLD.M.MAXEND statement 94
NLD.M.NETID statement 94
NLD.M.PDDNM statement 95
NLD.M.PFMEM statement 95
NLD.M.PIUTNP statement 96
NLD.M.PIUTSIZE statement 96
NLD.M.PURGE statement 96
NLD.M.RETRY statement 97
NLD.M.RTDASD statement 97
NLD.M.RTMDISP statement 98
NLD.M.RTM statement 98
NLD.M.RTDASD statement 97
NLD.M.RTMDISP statement 98
NLD.M.SAW statement 99
NLD.M.SAWN statement 99
NLD.M.SAWSIZE statement 100
NLD.M.SDDNM statement 100
NLD.M.SESSIONS statement 100
NLD.M.TRACEGW statement 101
NLD.M.TRACEUNIT statement 101
NLD.M.TRACESC statement 102
NMCstatus statement 102
NPDA.ALERTFWD statement 121
NPDA.ALERTLOG statement 123
NPDA.ALTINFFP statement 124
NPDA.ATT_LOCATE statement 125
NPDA.AUTHORATE statement 126
NPDA.DSRBO statement 127
NPDA.DSRBU statement 127
NPDA.JITHRESH statement 128
NPDA.LQTHRESH statement 129
NPDA.MACRF statement 129
NPDA.MDSIND statement 130
NPDA.NPDA.ERR_RATE
statement 127
NPDA.NPDA.DDDNM statement 131
NPDA.NPDA.NOTIFY statement 131
NPDA.R (Ratio) statement 132
NPDA.RATE statement 133
NPDA.REPORTS statement 134
NPDA.SDDNM statement 135
NPDA.TECROUTE statement 136
NPDA.W (Wrap) statement 136
NPDA.W (Wrap) statement 136
NQNSUP 139
NRM.CMODE statement 139
NRM.CONNINT statement 140
NRM.CONNRETRY statement 140
NRM.HBRETRY statement 141
NRM.HOSTDEST statement 141
NRM.PORT statement 142
NRM.RESET statement 142
NRM.SAMPLERATE statement 143
NRM.STATUS statement 143
NRM.TYPE statement 144
NRM.XCLDOM statement 144
NRM.XCLSRC statement 145
NRM.XCLTASKN statement 146
NRM.XCLTASKT statement 146
OpDiPrefix statement 149
POLICY statement 157
RMRODMUser statement 163
RMTINIT statement 166
RMTSYS statement 169
RDOMname statement 170
RRD statement 170
RSH statement 171
RIT statement 165, 172
SECOPTS.AUTHCHK statement 172
SECOPTS.CMDAUTH statement 173
SECOPTS.OPSEC statement 175
SECOPTS.OPSAPN statement 179
SECOPTS.SPANAUTH statement 180
SECOPTS.SPANCHK statement 181
SECOPTS.SPANAUTH statement 181
SECOPTS.WEBAUTH statement 181
SECOPTS.WEBAUTH statement 181
SSIDname statement 184
styleMsg statement 188
SuppChar statement 189
TAME statement 189
TASK statement 190
TCPname statement 192
TCPserver statement 192
TOWER statement 200
TRAGE statement 200
transMemStatement statement 202
### GMFHS initialization statements (continued)
- LCON-MAX-QUEUE-OPERIF statement 363
- LCON-MAX-QUEUE-RCMGR statement 363
- LCON-MAX-QUEUE-RTMGR statement 364
- LCON-MAX-QUEUE-VIEWMGR statement 364
- LCON-MAX-QUEUE-VSTATMGR statement 365
- LCON-NCC-RETLY-LIMIT statement 366
- LCON-NCC-RSC-LIMIT statement 366
- LCON-NMG-POLL-INTERVAL statement 367
- LCON-OPERATOR-CMD-AUDIT statement 368
- LCON-REPORT-UNKNOWN-STATUS statement 368
- LCON-SNATM-TIMEOUT statement 369
- LCON-OPERATOR-CMD-AUDIT statement 369
- LCON-OPERATOR-CMD-AUDIT statement 370
- LEVEL statement 370
- PRINTPDU38 statement 371
- RODMID statement 372
- RODMNAME statement 373
- STORAGE statement 373
- TASK statement 374
- TRACe statement 374
- TRACeBYTES statement 375
- TRACePAGES statement 376
- TYPE statement 376
- GMFHS time out 369
- GMTOFFSET statement 353

### H
- hardcopy printers, defining 51
- HARDCOPY statement 51
- hardware monitor
  - BER data 131
  - data display 125
  - database
    - event logging 124, 133
    - statistic logging 124
  - wrap counts, specifying 136
  - focal point nodes 37
  - NPDA.LQTHRESH statement 131
  - report record logging 134
- HCL keyword, PROFILE statement 160
- held area, command facility panel 55
- held messages, command facility panel 52
- HELD statement 52
- help panels 30
- high performance transport conversations 154
- highlight operands 59
- HLL statement 53
- HOLDPCNT statement 55
- HOSTNAME statement 317

### IC
- IC keyword
  - PROFILE statement 160
  - identifier, operator 150
- idpFarms statement 57
- IGRLSUP keyword
  - CMDMDL statement 28
- IHSAEVENT job 305
- IMDAREA statement 58
- immediate messages, command facility panel 58
- impulse hits threshold 128
- INCLUDE (%INCLUDE) statement 59
- INDENT statement 61
- indentation characteristics for messages
  - multiline 79
  - single text 61
- inform policy member 295
- CONTACT 299
- GROUP 298
- INFORM 298
- SETUP 295
- information, accessibility xiii
- information, disability xiii
- INIT keyword, TASK statement 190
- INIT statement 61
- INITFAIL keyword, KCLASS statement 66
- initial
  - ratio values 132
  - span of control 63
- initial status of resources 147
- initialization
  - CNMTAMEL task 14
  - command module, loading at 27
  - keywords
    - DST 41
    - NetView sequential log 72
    - network log 72
    - trace log 72
    - VPDTASK 205
  - record logging at 134
  - initialization type, remote processing 166
  - installation exits 70
  - INSTALLOPT 242
  - inStore statement 62
  - intensity operands 59
  - interconnected network 34, 73, 80
  - IQ_QUEUE_THRESHOLD statement 336
- IP_address keyword, RMTSYN statement 169
- IPC task message queue 359
- IPCONN 243
- IPHOST 245
- IPINFC 246
- IPLOG statement 63
- IPNAMESERV 248
- IPORT 250
- IPRUTER 253
- IPRN3270 255
- IRMGR task message queue 360


**J**

Japanese character translation 202, 203
JAPANESE statement 354
JesJobLog statement 64

**K**

kanji, character set 202, 203
KCLASS keyword, MAPSESS statement 77
KCLASS statement 64
keep class defining 64
mapping sessions into 64, 76
keep class definitions 90
KEEPPIU keyword
KCLASS statement 67
KCLASS keyword, MAPSESS statement 77
JesJobLog statement 64
JAPANESE statement 354
Japanese character translation 202, 203
J

**L**

labels 6
last line, command facility panel 68
LASTLINE statement 68
LBUILD macroinstruction 185
LCON-AGG-BUNDLE-INTERVAL statement 354
LCON-AGGRST-REQUIRED statement 355
LCON-AIP-RESET-INTERVAL statement 355
LCON-ALERT-CMD-TIMEOUT statement 356
LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement 356
LCON-EXCHANGE-BUFFER-INTERVAL statement 357
LCON-HEX-SUBVECTOR-DISPLAY statement 357
LCON-MAX-LOCATE-RESOURCE-VIEWS statement 358
LCON-MAX-QUEUE-DIBSERVER statement 358
LCON-MAX-QUEUE-EVENTMGR statement 359
LCON-MAX-QUEUE-IPC statement 359
LCON-MAX-QUEUE-IRMGR statement 360
LCON-MAX-QUEUE-MAINTASK statement 361
LCON-MAX-QUEUE-NETCMD statement 361
LCON-MAX-QUEUE-NETCON statement 362
LCON-MAX-QUEUE-OPERIF statement 363
LCON-MAX-QUEUE-RCMGR statement 363
LCON-MAX-QUEUE-RTMGR statement 364
LCON-MAX-QUEUE-VIEWMGR statement 364
LCON-MAX-QUEUE-VSTATMGR statement 365
LCON-NCC-RETRY-LIMIT statement 366
LCON-NCC-RSC-LIMIT statement 366
LCON-NMG-POLL-INTERVAL statement 367
LCON-OPERATOR-CMD-AUDIT statement 368
LCON-REPORT-UNKNOWN-STATUS statement 368
LCON-SNATM-TIMEOUT statement 369
LCON-STATUS-DELAY-MAX statement 369
LCON-STATUS-DELAY-TIME statement 370
leased lines 128
LEVEL statement 370
lines of unequal speeds 283
LOADEXIT statement 69
loading installation exits 69
local shared resources 45
locate resource request 369
lock indicator, command facility panel 70
LOCKIND statement 70
LOG_LEVEL statement 337
logging vital product data 205
logical unit (LU) name
CNMTARG statement 74
names
  defining 74
  translating 73
  session authorization 74, 75
logical units 93
LOGINIT statement 72
logon
  mode names 80
  logon authority 175
logon panel 22
LOGONPW statement 73
LPDA-2 modems 128, 129
LSTHRESH 257
LU communications 154
  full network qualified names 139
  tracing 101
LU keyword
  CNMTARG statement 74
  RMTSEC statement 168
LU statement 73
LUC.CNMTARG statement 74
LUC.CTL statement 75
LUC.MAXSESS statement 75
LUC.PERSIST statement 75
LUC statement 74
M
M (Maximum) statement 75
MACRF 93, 129
MACRF keyword, DISTINIT statement 45
MAINTASK task message queue 361
  manuals
  feedback xii
  online xii
  ordering xii
MAIPSS statement 76
MAX CHUNK statement 337
MAX SEGMENT NUM 338
MAX_WINDOW_NUM statement 339
MCON statement 78
MDMCNTL command 128, 129
MEM keyword, TASK statement 190
memStore statement 79
message
  authorized receiver 16
  BNJ018I 129
  BNJ045I 134
  buffer, parsing 28
  DS1151I 55
  indicator
    classes 10, 47
    defining alarm attribute 10
    defining color attribute 9
    defining highlighting attribute 10
    prefix information 158
  unsolicited 15
Message Classes 265
  messages
    DS1531I 50
    DS1546I 50
    DS1547I 50
    DS1559I 50
  translation 202
method
  object independent 355
  resource traits change 357
migration values for MVS interfaces 82
MLINDENT statement 79
MLOG_LEVEL statement 339
MOD keyword
  CMDMDL statement 25
  TASK statement 190
MODE keyword, RMTINIT statement 166
mode names, logon 80
MODE statement 80
modems, LPDA-2 128
MONIT 258
  MONIT STOP command 147
  MONITOR control file entry 259
  MONOFF command 147
MS transport 206
MSGCFG statement 318
MSGIFAC statement 83
MSRECVR keyword, AUTH statement 16
MSU_CORR keyword, W (Wrap) statement 137
MSUs, automating 126
MTRACE_TYPE statement 340
MVS
  extended console support 81
  MVS command
    console name 161
  MVS message interface 83
  MVSPARM.MGIFAC statement 83
  MVSPARM statement 81, 82
N
NAME keyword
  PARTNER statement 154
NCP activation
  gateway tracing 101
NCPCRECOV (SNA) 260
NET keyword, RMTSEC statement 168
NETCMD task message queue 361
NETCMS class 173
NETCON task message queue 362
NETCONV IP statement 151
NetFinity 295
NETID keyword
  PARTNER statement 154
  netid keyword, RMTSYN statement 169
NetID statement 84
NetSP Single Logon Coordinator 177
NetView
  application name 86
  banner 22
  defaults 35
  samples, location 1
  sequential log 72
NetView 3270 management console definition statement
  OPERID 151
  PORT 157
  PROTOCOL 162
  SOCKETS 183
  TCPEXAME 191
NetView Web Server
  definition statement
    PORT 157
    SOCKETS 183
    TCPEXAME 191
NetViewAlertReceiver statement 319
network
  interconnected 34
  log, initial keywords 72
  name, defining 152
  status updates, receiving 148
Network Security Program 177
NEWLINE keyword, INDENT statement 61
NGMADDM keyword, AUTH statement 17
NGMFCDM keyword, AUTH statement 17
NGMFVSPN keyword, AUTH statement 17
NLDM.AMLUTDLY statement 84
NLDM.AUTHDOM statement 85
PPI statement 321
PRI keyword
KCLASS statement 67
MAPSESS statement 77
TASK statement 190
PRIMARY_HEAP_SIZE statement 341
PRIMARY keyword, DEFFOCPT statement 39
PRINTPDU38 statement 371
processing, ending 47
PROFILE statement
CONSTATE keyword 161
HCL keyword 161
IC keyword 161
overview 160
PROFILEN statement 161
program-to-program interface command receiver, defining 29
programmable network access support registration table 162
PROTOCOL statement 162
publications
feedback xii
online xii
ordering xii
PUCOUNT statement 162
PURGE command 35

Q
queue
Dbserver task message 358
EVENTMGR task message 359
IPC task message 359
IRMGR task message 360
MAINTASK task message 361
NETCMD task message 361
NETCON task message 362
OPERPARM segment 82
ratio, error-to-traffic (E/T) 132
RCM
RCMRODMUser 163
RCMGR task message queue 363
RCMRODMUser statement 163
receiving network status updates 148
RECOVERY control file entry 269
remote communications 169
operator, authorize or block 167
processing initialization type 166
RENAME keyword, KCLASS statement 67
REP statement 163
report record logging 134
REPORTS statement 134
RES_CORR keyword, W (Wrap) statement 138
RES keyword, CMDMDL statement 27
RESIDENT control file entries in the control file 274
resource activation by status monitor 75
Resource Trait subtask 364
resource traits change method, time interval 357
response time monitor measurement keywords 155
RESUME keyword, LOGINIT statement 72
RETRY keyword
DISTINIT statement 46
NLDM.CDRMDEF statement 86
RETRY statement 165
RetryInterval statement 322
REXEC server 165
REXEC server files definition statement
PROTOCOL 162
REXEC statement 165
RMTINIT statement 166
RMTOP keyword, RMTSEC statement 168
RMTSEC statement 167
RMTSYN statement 169
RODM
access and control task 163
automation task definition statement
CMDRCVR 29
REP 163
TASK 191
definition statement 329
ASYNC_TASKS 329
CELL_POOLS 329
CHARACTER_VALIDATION 331
CHECKPOINT_FUNCTION 332
CONCURRENT_USERS 333
CONNECT_VIOLATION_MESSAGE statement 333
DUMP_FOR_BAD_USER_DATA 334
DUMP_LIMIT 334
DUMP_SCOPE 335
EXTEND_HEAP_SIZE 336
IO_QUEUE_THRESHOLD 336
LOG_LEVEL 337
MAX_CHUNK 337
MAX_SEGMENT_NUM 338
MAX_WINDOW_NUM 339
MLOG_LEVEL 339
MTRACE_TYPE 340
PLI_ISA 340
PRIMARY_HEAP_SIZE 341
QUIES_WAIT_TASK_TIME 342
QUIES_WAIT_USER_TIME statement 342
RODM (continued)
definition statement (continued)
WINDOW_PockETS 348
distribute commands 191
RODMID statement 372
RODMName statement 170
RODMNAME statement 373
RRD statement 170
RRS interfaces 34
RSH server definitions 171
RSH server files definition statement
PROTOCOL 162
RSH statement 171
RTDEF keyword, PCLASS statement 156
RTM data 98
performance class definitions 95
recording 88
RTMDATA keyword, KCLASS statement 66
RTT statement 172
running AON and SA/390 in the same NetView domain 240
S
SAF refresh keyword, RMTINIT statement 166
Sample Policy Definitions 103
samples, location in NetView program 1
SAW keyword
KCLASS statement 68
SBSDN keyword, DISTINIT statement 43
screen format statements
%INCLUDE statement 59
ACTION statement 12
CMDLINE statement 24
COLUMNHEAD statement 31
HELD statement 52
HOLDPCNT statement 55
HOLDWARN statement 55
IMDAREA statement 58
INDENT statement 61
LASTLINE statement 68
LOCKIND statement 70
MLINDENT statement 79
NOPREFIX statement 117
NORMAL statement 117
NORMQMMax statement 118
PREFIX statement 158
TITLE statement 192
TITLEDATE statement 194
TITLEDOMID statement 195
TITLEOID statement 196
TITLESTAT statement 197
TITLETIME statement 199
SDDNM keyword, DISTINIT statement 44
screen format statements
%INCLUDE statement 59
ACTION statement 12
CMDLINE statement 24
COLUMNHEAD statement 31
HELD statement 52
HOLDPCNT statement 55
HOLDWARN statement 55
IMDAREA statement 58
INDENT statement 61
LASTLINE statement 68
LOCKIND statement 70
MLINDENT statement 79
NOPREFIX statement 117
NORMAL statement 117
NORMQMMax statement 118
PREFIX statement 158
TITLE statement 192
TITLEDATE statement 194
TITLEDOMID statement 195
TITLEOID statement 196
TITLESTAT statement 197
TITLETIME statement 199
SDDNM keyword, DISTINIT statement 44
screen format statements
%INCLUDE statement 59
ACTION statement 12
CMDLINE statement 24
COLUMNHEAD statement 31
HELD statement 52
HOLDPCNT statement 55
HOLDWARN statement 55
IMDAREA statement 58
INDENT statement 61
LASTLINE statement 68
LOCKIND statement 70
MLINDENT statement 79
NOPREFIX statement 117
NORMAL statement 117
NORMQMMax statement 118
PREFIX statement 158
TITLE statement 192
TITLEDATE statement 194
TITLEDOMID statement 195
TITLEOID statement 196
TITLESTAT statement 197
TITLETIME statement 199
Index 387
SECOPTS.CMDAUTH statement 173
SECOPTS.OPERSEC statement 175
SECOPTS.OPSPAN statement 179
SECOPTS.SPANAUTH statement 180
SECOPTS.SPANCHK statement 181
SECOPTS.SURROGAT statement 181
SECOPTS.WEBAUTH statement 181
security
command authority 173
operator 175
span of control 180
security checking 34
SEGMENT_POCKETS statement 345
SENDMSG statement 182
ServerLocation statement 322
ServerPort statement 323
service, class of 34
SESSFAIL keyword, KCLASS statement 66
session
CNM data transfer 75
LU, authorization 75
monitor
conversations 46, 86
session data retrieval 86
parallel 75
persistent 74
TAF 47
SESSION (SNA) statement for sessions monitored by AON/SNA 274
session awareness data 99, 100
session configuration data authorization 86
session monitor
availability data 92, 100
cross-domain session 87
DASD session wrapping 91
data base tuning 96
ER data 97
explicit routes 89
external log 92
flow control data request 89
gateway tracing 101
initial conversation 97
keep class 90
logical units, number 93
LU tracing 101
PIU trace data 88, 90, 94
recording a session to database 88, 89
response time collection period 91
RTM data 88, 98
SSCP tracing 102
SESSNORM keyword, KCLASS statement 67
shortcut keys, keyboard xiii
SMF log 92
SNA keyword, RMTSYNC statement 170
SNA-MDS keyword, ALERTFWD statement 122
SNA Topology Manager
response to GMFHS 369
SNAPRQ keyword, VPDINIT statement 205
SNBU (SNA) 275
SNBUPOOL control file entry 278
SOCKETS keyword, RMTINIT statement 167
SOCKETS statement 183
span definitions 180
span of control 179
authorization 181
span of control 63
SPAN statement
definition 183
SPACK keyword, DSTINIT statement 44
SRATIO command 132, 136
SSB_CHAIN statement 346
SSCP
cross-domain, name 86
session awareness data for 64
tracing 102
SSIDname statement 184
ST keyword, W (Wrap) statement 137
START command 190
START SPAN command 179, 183
START TASK command 170
statements, location in samples 1
statistic record recording 124
STOPT
default descriptions 187
statement 185
status indicators, command facility panel 197
status monitor
as secondary resource 148
displaying commands or command lists 23
functions 185
initialization message for resources 149
MONIT function 75
reactivating nodes 147
STOP command 63
STORAGE statement 373
styleMsg statement 188
SUBSYSTEM control file entry for NVAS subsystem interface with SNA help desk 279
subsystem interface 184
SUBSYSTEM statement 188
subtask, Resource Trait 364
SuppChar statement 189
suppression character 189
SWITCH command 35
switching traffic load back to the primary line 283
SWRAP command 132, 136
synonyms, defining command lists 30
commands 30
keywords 153
syntax conventions 6
syslog server 63
system authorization facility product APPL class 176
T
TAF sessions 47
TALRTCFG statement 324
TAMEL statement 189
TARGOP keyword, RMTSEC statement 168
TASK statement 190, 374
tasks, defining 190
TCP/IP
connectivity
CNMTAMEL 189
DSIPLAN 63
DSIREXCF 165
DSIRSH 171
DSIRTR 172
DSITCPPIP 78
DSIBMSTK 207
TCP/IP network resources to AON/TCP 282
TCP390 control file entry 279
TCPANAME statement 191
TCPname statement 192
TCPserver statement 192
terminal access facility 50
TestMode statement 325
TCSWITCH control file entry 283
THRESHOLDS, setting 284
time-out value 87
TIMER control file entries 287
title line
command facility panel 192
date, command facility panel 194
time, command facility panel 199
TITLE statement 192
TILEDATE statement 194
TILEDOMID statement 195
TITLEPID statement 196
TITLSTAT statement 197
TITLTIME statement 199
tivoli Customer Support xiii
TOWER statement 200
TP keyword, MAPSESS statement 78
TRACDATA keyword, KCLASS statement 67
trace
log 72
VSAM database performance, warning 375
trace data
authorization 86
TRACE initialization parameters 200
TRACER statement 200, 325, 374
TRACEBYTES statement 375
TRACEPAGES statement 376
translation
alias name 13
character 203
messages 202
transMember statement 202
TRANSPARENT_CHECKPOINT statement 346
transTbl statement 203
TSO user ID surrogate 181
TSOENV 291
TYPE keyword
CMDDMDL statement 25
DEFFOCPT statement 37, 38
TYPE statement 376
UNSOL keyword, DSTINIT statement 43
unsolicited messages, routing to an operator 15
UsePortMapper statement 326
UserStatus field 355

VALCLASS statement 203
VBUILD macro 185
VBV statement 204
view updates 357
VIEWMGR task message queue 364
Visual BLDVIEWS initialization parameters 204
vital product data, logging 205
VPDINIT statement 205
VPDREQ keyword, VPDINIT statement 205
VPDSTOR keyword, VPDINIT statement 206
VPDTASK subtask 205
VPDWAIT keyword, VPDINIT statement 205
VR keyword, MAPSESS statement 78
VSAM database trace, warning 375
file, mapping sessions to 66
services 43
VSTATMGR task message queue 365
VTAM CDRSC definition statement 73
spans 183
VTAMCP statement 206

WAIT_ALLOC_TIME statement 347
web access 207
web server
authorization 181
WEB_SERVER statement 152
WEB statement 207
WINDOW_CHKPT_TIME statement 348
WINDOW_POCKETS statement 348
WLM statement 207
WLM support 207
workload management 207
wrap count
initial 136
WRAPCNT keyword, NPDA.ALCACHE statement 120

X25MONIT control file entry 291
XITBN keyword, DSTINIT statement 44
XITBO keyword, DSTINIT statement 44
XITCI keyword, DSTINIT statement 45
XITCM keyword, DSTINIT statement 44
XITCO keyword, DSTINIT statement 45
XITDI keyword, DSTINIT statement 44
XITVI keyword, DSTINIT statement 45
XITVN keyword, DSTINIT statement 45
XITVO keyword, DSTINIT statement 45
XITXL keyword, DSTINIT statement 45
XLO keyword, REPORTS statement 135