

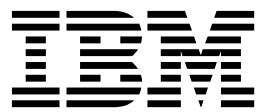
IBM Tivoli NetView for z/OS
Version 6 Release 2 Modification 1

Administration Reference



IBM Tivoli NetView for z/OS
Version 6 Release 2 Modification 1

Administration Reference



Note

Before using this information and the product it supports, read the information in "Notices" on page 603.

This edition applies to version 6, release 2, modification 1 of IBM Tivoli NetView for z/OS (product number 5697-NV6) and to all subsequent versions, releases, and modifications until otherwise indicated in new editions.

This edition replaces SC27-2869-05.

© Copyright IBM Corporation 1997, 2015.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this publication	xv
Intended audience	xv
Publications	xv
IBM Tivoli NetView for z/OS library	xv
Related publications	xvii
Accessing terminology online	xvii
Using NetView for z/OS online help	xviii
Accessing publications online	xviii
Ordering publications	xviii
Accessibility	xviii
Service Management Connect	xix
Tivoli technical training	xix
Tivoli user groups	xix
Downloads	xix
Support information	xx
Conventions used in this publication	xx
Revision codes	xx
Typeface conventions	xx
Operating system-dependent variables and paths	xxi
Syntax diagrams	xxi
Chapter 1. Location of Statements and Samples	1
How to Use the NetView Definition Statement Reference	1
Statement Formats	4
Syntax Conventions for Definition Statements	4
Chapter 2. CNMSTYLE Initialization Statements	7
ACBpassword	7
AlertRcvName	8
ARCHIVE.ACCESSDELAY	8
ARCHIVE.BROWSE.DATASPACES	9
ARCHIVE.HLQ	10
ARCHIVE.INDEX.BLOCKSIZE	11
ARCHIVE.INDEX.DATACLAS	11
ARCHIVE.INDEX.MGMTCLAS	12
ARCHIVE.INDEX.SPACE	14
ARCHIVE.INDEX.STORCLAS	15
ARCHIVE.INDEX.UNIT	16
ARCHIVE.INDEX.VOLUME	17
ARCHIVE.MESSAGE.BLOCKSIZE	18
ARCHIVE.MESSAGE.DATACLAS	19
ARCHIVE.MESSAGE.MGMTCLAS	20
ARCHIVE.MESSAGE.SPACE	21
ARCHIVE.MESSAGE.STORCLAS	22
ARCHIVE.MESSAGE.UNIT	23
ARCHIVE.MESSAGE VOLUMES	24
ARCHIVE.WRITE	25
ASSIGN	26
AUTOCMD	27
AUTOTASK	28
auxInitCmd	29
banner	30
CCDEF	30
CNMI	31
COMMON. <i>variable_name</i>	31

COMMON.CNMIP.DNSTimeout	32
COMMON.CNMSNMP.MIBPATH	33
COMMON.CNMSNMP.MIBS	34
COMMON.CNMSNMP.port	35
COMMON.CNMSNMP.retries	35
COMMON.CNMSNMP.timeout.	35
COMMON.CNMSNMP.trap	36
COMMON.CNMTRAP.taskname.CONFIGFILE	36
COMMON.CNMTRAP.taskname.MAXTCPCONN	37
COMMON.CNMTRAP.taskname.STACKNAME	38
COMMON.CNMTRAP.taskname.TCPPORT	38
COMMON.CNMTRAP.taskname.TRACE	39
COMMON.CNMTRAP.taskname.UDPPORT	40
COMMON.DUIFHNAME	40
COMMON.DUIFPRC	41
COMMON.EKGHNAM	41
COMMON.EKGHPRC	42
COMMON.EZLIMITDELAY	42
COMMON.EZLIPTraceJCLWait	42
COMMON.EZLRMTIMER	43
COMMON.EZLsmtpHOSTNAME	43
COMMON.EZLsmtpNAME	44
COMMON.EZLTCPrACEwriter.	44
COMMON.EZLTCPpTRACEwriter	45
COMMON.EZLTRACE	45
COMMON.FLC_DEF_NETW_VIEW	46
COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS	46
COMMON.FLC_EXCEPTION_VIEW_FILE	47
COMMON.FLC_RODMAPPL	47
COMMON.FLC_RODMINT	48
COMMON.FLC_RODMNAME	48
COMMON.FLC_RODMRETRY	49
COMMON.FLC_RUNCMDRETRY.	49
COMMON.FLC_TCPNAME	49
COMMON.IPPORTMON.INTVL	50
COMMON.IPPORTMON.IPADD	50
COMMON.IPPORTMON.PORTNUM	51
COMMON.IPPORTMON.PORTTO	51
COMMON.PKTS.INSTANCE.POOL	52
COMMON.SMFVPD	52
COMMON.STACKFAMILY	53
COMMON.WAITTIME	53
COMMON.XDOMTIME	54
ConsFixed.	54
ConsMask	55
CORRELATION	56
DB2SEC	58
DEFAULTS	58
DISCOVERY.INTAPPL	59
DISCOVERY.INTINTERFACE	60
DISCOVERY.INTTELNET.	60
DISCOVERY.NetViewOnly	61
DISCOVERY.SNMP.	61
DLA.cmdb_ftp_server	62
DLA.debug	62
DLA.ftp_log_blocksize.	63
DLA.ftp_log_filename	63
DLA.ftp_log_space_pri	64
DLA.ftp_log_space_sec	64
DLA.ftp_log_unit	65
DLA.ftp_log_volume	66

DLA.ftp_max_xmit_tm	66
DLA.ftp_remote_dir	67
DLA.ftp_timeout	67
DLA.ftp_uid	68
DLA.is_second_level	68
DLA.statefile	69
DLA.tsouser	69
DLA.xml_blocksize	70
DLA.xml_filename	70
DLA.xml_space_pri.	71
DLA.xml_space_sec.	72
DLA.xml_unit	72
DLA.xml_volume	73
DOMAIN	73
DVIPA.Event.Delay.	74
DVIPA.INTDVTAD.	75
DVIPA.INTDVCONN	75
DVIPA.INTDVDEF	76
DVIPA.INTDVROUT	76
DVIPA.Mast.Disc.Delay	77
DVIPA.Mast.EMARf.Delay	77
DVIPA.STATS.DVIPA	78
DVIPA.STATS.Logto	79
DVIPA.STATS.PORT	80
DVIPA.STATS.Pri.MAXR	80
DVIPA.STATS.Sec.MAXR	81
DVIPA.STATS.TCPNAME	81
EMAAUTO	82
endcmd.AutoTask	83
endcmd.close.leeway	83
ENT.CONNCHECK.INT	84
ENT.GROUP, <i>groupname</i>	85
ENT.INT. <i>name</i>	86
ENT.SYSTEMS. <i>name</i>	87
function.autotask	88
GHB.TCPANAME	89
HARDCOPY	90
HLLENV	91
idleParms	92
IDS.Attack_Cmd.	93
IDS.Attack_CmdType	94
IDS.Auto_Intvl	95
IDS.Auto_Thresh	96
IDS.ClearStat_Day	96
IDS.ClearStat_Inform	97
IDS.ClearStat_Log	98
IDS.ClearStat_Log_File	99
IDS.ClearStat_Time	99
IDS.CONSOLEMSG	100
IDS.DSIPARM	100
IDS.Event_Inform	101
IDS.Event_Limit	102
IDS.Event_Log	102
IDS.Event_Log_File	103
IDS.Flood_Cmd	103
IDS.Flood_CmdType	104
IDS.probeid	105
IDS.Report_Cmd	106
IDS.Report_CmdType	107
IDS.Report_Inform	108
IDS.Report_Log	108

IDS.Report_Log_File	109
IDS.Scan_Cmd	109
IDS.Scan_CmdType	110
IDS.SYSLOG.FILENAME	111
IDS.SYSLOGMSG	112
IDS.TCP_Cmd	113
IDS.TCP_CmdType	113
IDS.UDP_Cmd	114
IDS.UDP_CmdType	115
INFORM.POLICY.MEMBER	116
INIT.DVIPASTATS	116
INIT.EMAAUTO	117
INIT.NRM	117
INIT.OPKT	118
INIT.PKTS	119
INIT.TCPCONN	119
INIT.TIMER	120
inStore	120
IPLOG	121
IPv6Env	122
verbose	122
JesJobLog	123
LOADEXIT	123
LOGONPW	125
LUC	125
memStore	127
MODIFY.TOWER	128
MSG.TECROUTE	129
MVSPARM.ActionDescCodes	130
MVSPARM.Cmd.Designator	131
MVSPARM.Cmd.Scope	132
MVSPARM.DEAUTH	132
MVSPARM.Msg.Automation	134
MVSPARM.Msg.Automation.MaxAge	134
MVSPARM.Msg.Automation.Oldtag	135
MVSPARM.OperRecvBrdcst	136
NACMD.DESTPPI	136
NACMD.INTCONINACT	137
NACMD.INTCONNACT	138
NACMD.INTHEALTH	138
NACMD.INTSESSACT	139
NACMD.LCLPPIRV	139
NACMD.OPID	140
NACMD.PERSIST	141
NACMD.ROWSxxx	142
NACMD.SUBNODE	143
NACMD.WAITSECS	144
NetID	145
NLDM.AMLUTDLY	145
NLDM.AUTHDOM	146
NLDM.AUTHORIZ	147
NLDM.CDRMDEF	148
NLDM.CDTIME	149
NLDM.DRDELAY	150
NLDM.DSRBO	150
NLDM.ERCOUNT	151
NLDM.FCTIME	151
NLDM.KEEPDISC	152
NLDM.KEEPMEM	152
NLDM.KEEPPIU	153
NLDM.KEEPRTM	153

NLDM.KEEPSESS	154
NLDM.LOG	154
NLDM.LUCOUNT	155
NLDM.MACRF	156
NLDM.MAXEND	157
NLDM.NETID	157
NLDM.PDDNM	158
NLDM.PERFMEM	158
NLDM.PEXLST xx	159
NLDM.PIUTNUM	160
NLDM.PIUTSIZE	160
NLDM.RETRY	161
NLDM.RTDASD	161
NLDM.RTM	162
NLDM.RTMDISP	162
NLDM.SAW	163
NLDM.SAWNAM	163
NLDM.SAWSIZE	164
NLDM.SDDNM	164
NLDM.SESSIONMAX	165
NLDM.SESSTATS	165
NLDM.TRACEGW	166
NLDM.TRACELU	166
NLDM.TRACESC	167
NMCstatus.errorDSN	168
NPDA.ALCACHE	168
NPDA.ALERTFWD	170
NPDA.ALERTLOG	172
NPDA.ALRTINFP	173
NPDA.ALTER_ALERT	174
NPDA.AUTORATE	175
NPDA.DSRBO	176
NPDA.DSRBU	177
NPDA.ERR_RATE	177
NPDA.MACRF	178
NPDA.MDSIND	179
NPDA.PDDNM	179
NPDA.PDFILTER	180
NPDA.PRELOAD_BER	180
NPDA.R(Ratio)	181
NPDA.RATE	182
NPDA.REPORTS	183
NPDA.SDDNM	184
NPDA.TECROUTE	185
NPDA.W (Wrap)	185
NQNSUP.USE	188
NRM.CMODE	189
NRM.CONNINT	190
NRM.CONNRETRY	190
NRM.HBRETRY	191
NRM.HOSTDEST	191
NRM.IPV6SRC	192
NRM.PORT	193
NRM.RESET	194
NRM.SAMPLERATE	194
NRM.STATUS	195
NRM.TYPE	196
NRM.XCLDOM	196
NRM.XCLSRC	197
NRM.XCLTASKN	198
NRM.XCLTASKT	198

NVROLE	199
NVSP. <i>srvrname</i> .NUMTHRDS	200
NVSP. <i>srvrname</i> .PDS	200
NVSP. <i>srvrname</i> .PORT	201
NVSP. <i>srvrname</i> .SECURE	202
NVSP. <i>srvrname</i> .TRC	203
NVSP. <i>srvrname</i> .WAIT	204
OpDsPrefix	204
OPKT.STORAGE	205
PKTS.STORAGE	206
POLICY	206
PWD	207
RCMRODMUser	208
RESTORE.CGLOBAL	208
RExec.PORT	209
RExec.SOCKETS	210
RExec.TCPNAME	211
REXX.CMDENV.name	211
REXX.FUNCPKGLIST.LCL.name	212
REXX.FUNCPKGLIST.SYS.name	213
REXX.FUNCPKGLIST.USR.name	213
RMTALIAS	214
RMTINIT.IP	216
RMTINIT.PORT	216
RMTINIT.SAFrefresh	217
RMTINIT.SECOPT	217
RMTINIT.SOCKETS	218
RMTINIT.SNA	218
RMTINIT.TCPNAME	219
RMTINIT.KEEPALIVE	219
RMTSYN	220
RODMname	222
ROUTECDE	222
RRD	223
RSH.PORT	224
RSH.SOCKETS	224
RTNDEF.BASE.COLLECTOR.APPL	225
RTNDEF.BASE.COLLECTOR.STACK	225
RTNDEF.BASE.COLLECTOR.SYSTEM	226
RTNDEF.BASE.HB.APPL	226
RTNDEF.BASE.MEMBER	226
RTNDEF.BASE.PROCESSOR.APPL	227
RTNDEF.BASE.PROCESSOR.STACK	227
RTNDEF.BASE.PROCESSOR.SYSTEM	228
RTNDEF.BASE.PROVIDER	228
RTNDEF.COLLECTOR.APPL.NETVIEW	228
RTNDEF.PROCESSOR.STACK	229
RTT.PORT	229
RTT.SOCKETS	230
RTT.TCPNAME	230
SECOPTS.AUTHCHK	231
SECOPTS.CMDAUTH	231
SECOPTS.OPERSEC	234
SECOPTS.OPSPAN	238
SECOPTS.RMTAUTH	239
SECOPTS.SARESAUTH	240
SECOPTS.SPANAUTH	241
SECOPTS.SPANCHK	241
SECOPTS.SURROGAT	242
SECOPTS.WEBAUTH	242
SSI.PPI	243

SSI.ProcString	243
SSI.ReviseTable	244
styleMsg	245
styleMsg.wrap	245
stylevar	246
SuppChar	247
TAMEL.CONV.sysdef	248
TAMEL.PORT	249
TAMEL.SOCKETS	249
TAMEL.TCPANAME	250
TAMEL.TTLS	250
TAMEL.USETCPPIP	251
TASK.taskname.INIT	252
TASK.taskname.MEM	253
TASK.taskname.MOD	253
TASK.taskname.PRI	254
TCPCONN.DASD.stackname	255
TCPCONN.DSRBO	256
TCPCONN.ERRMSG.THROTTLE	256
TCPCONN.GTF.stackname	257
TCPCONN.HASHSIZE.stackname	257
TCPCONN.KEEP.stackname	258
TCPCONN.MACRF	259
TCPCONN.PDDNM	259
TCPCONN.ROWSA.stackname	260
TCPCONN.ROWSD.stackname	261
TCPCONN.SDDNM	261
TCPname.	262
TCPserver	262
TOWER	263
TRACE.MODE	266
TRACE.MODFILT	267
TRACE.MONOPER	268
TRACE.OPTION	269
TRACE.SAFopt.	270
TRACE.SAFrc	271
TRACE.SIZE	271
TRACE.TASK	272
transMember	272
transTbl	273
VTAMCP.USE	273
WBSORDER.	274
WEB	275
webmenu.	276
WEBSITE.	278
WINDOW.EXEC	279
WLM.SubSystemName	279
XCF.GROUPNUM	279
XCF.MASTDVIPA	280
XCF.PROCSTR <i>nn</i>	281
XCF.RANK	282
XCF.TAKEOVER.CLIST	282
XCF.TAKEOVER.CONVIP <i>nn</i>	283
XCF.TAKEOVER.CONVSNA <i>nn</i>	284
XCF.TAKEOVERDELAY	284
XCF.TAKEOVER.DURATION	285
XCF.TAKEOVER.INITWAIT	286
XCF.TAKEOVER.NETCONVS	286
Chapter 3. NetView Definition Statements	289
A (Alert)	289

ACCESS	291
ACTION	292
ALIASMEM	293
AMELINIT	294
AUTH	295
C (Command List)	300
CMDCLASS	301
CMDDEF	301
CMDLINE	308
CMDMDL	309
CMDRCVR	309
CMDSYN	310
COLUMNHEAD	310
COMNTESC	312
COS	312
DBFULL	313
DEFENTPT	313
DEFFOCPT	314
DEFFOCPT Statement in DSICRTTD	315
DEFFOCPT Statement in DS16INIT	316
DOMAINS	317
DSTINIT	318
END	324
F (Filter)	324
HELD	327
HOLDPCNT	329
HOLDWARN	330
IMDAREA	331
%INCLUDE	333
INDENT	335
IS PAN	336
KCLASS	337
KEYCLASS	341
LASTLINE	341
LOCKIND	343
LOGINIT	344
LU	345
M (Maximum)	346
MAPSESS	347
MLINDENT	349
MODE	350
MVS	351
NETCONV_IP	352
NMCSTATUS (Control File Entry)	353
NOPREFIX	367
NORMAL	367
NORMQMAX	369
O MONIT	370
O RESET	370
O SECSTAT	371
O SENDMSG	372
OPCLASS	372
OPERATOR	373
ORIGNET	374
PARMSYN	375
PARTNER	375
PCLASS	376
PORT	379
PREFIX	379
PROFILE	382
PROFILEN	383

PROTOCOL	384
REP	384
RETRY	386
RMTSEC	386
SENDMSG	388
SOCKETS	389
SPAN	390
STATOPT	391
SUBSYSTEM	395
TASK (DSIQTSKI)	395
TCPNAME	395
TITLE	396
TITLEDATE	398
TITLEDOMID	399
TITLEOPID	401
TITLESTAT	402
TITLETIME	404
VALCLASS	405
VPDINIT	406
WEB_SERVER	407
WINDOW	408
Chapter 4. Policy File Definitions	409
Policy File Entries	409
ACTMON	409
ADJNETV	414
AUTOOPS	415
CDLOG	419
CPCPSESS (SNA)	421
DDFGENERIC	422
DDFGROUP	424
ENVIRON AIP	424
ENVIRON DDF	425
ENVIRON EXIT	427
ENVIRON RACF	429
ENVIRON SETUP	431
ENVIRON TIMEOUT	434
EZLTLOG	435
FORWARD FOCALPT	436
FULLSESS	437
GATEWAY	438
INSTALLOPT	440
IPCONN	442
IPHOST	444
IPINFC	446
IPNAMESERV	448
IPPORT	450
IPROUTER	453
IPTELNET	455
IPTN3270	456
LSTHRESH	459
MONIT	460
MONITOR	461
NCPRECOV (SNA)	462
NOTIFY	465
NTFYOP	466
RECOVERY	469
RESIDENT	474
SESSION	475
SUBSYSTEM (SNA)	475
TCP390	476

TGSWITCH (SNA)	481
THRESHOLDS	482
TIMER	486
TSOSERV.	489
X25MONIT (SNA).	490
Chapter 5. Inform Policy Member	493
SETUP	497
GROUP	499
INFORM	499
CONTACT	500
Chapter 6. Event/Automation Service Definition Statements	501
AdapterCdsFile.	502
AdapterFmtFile.	503
ALRTCFG	504
ALRTTCFG	505
BufEvtMaxSize	506
BufEvtNegRespLimit	506
BufEvtPath	507
BufEvtRdBlklen	508
BufEvtShrinkSize	509
BufferEvents.	510
BufferEventsLimit	511
BufferFlushRate	511
CALRTCFG	512
CMSGCFG	513
Community	514
ConnectionMode	514
Enterpriseoid	515
ERCVCFG	516
EventMaxSize	517
FallbackValue	517
Filter	518
FilterCache	519
FilterMode	520
Hostname	521
MSGCFG.	521
NetViewAlertReceiver	522
NOSTART	523
OUTPUT.	524
PortNumber.	525
PPI.	525
RetryInterval	526
ServerFallback	527
ServerLocation	528
ServerPort	529
TALRTCFG	530
TestMode.	531
TRACE	531
TruncateSV31s	533
UsePortMapper.	534
Chapter 7. Correlation Service Definition Statements	535
CLIENTPORT	535
CLIENTLISTPORT	535
JLOG	536
LCLPORT	537
NVHOST.	537
NVLOCATE.	538

NVPORT	538
PURGETIMER	539
QUELIMIT	539
RETRYTIME	540
RULEFILE	540
Chapter 8. Resource Object Data Manager Definition Statements	541
ASYNC_TASKS	541
CELL_POOLS	541
CHARACTER_VALIDATION	543
CHECKPOINT_FUNCTION	544
CONCURRENT_USERS	545
CONNECT_VIOLATION_MESSAGE	546
DUMP_FOR_BAD_USER_DATA	546
DUMP_LIMIT	547
DUMP_SCOPE	548
EXTEND_HEAP_SIZE	549
IO_QUEUE_THRESHOLD	549
LOG_LEVEL	550
MAX_CHUNK	550
MAX_SEGMENT_NUM	551
MAX_WINDOW_NUM	551
MLOG_LEVEL	552
MTRACE_TYPE	553
PLI_ISA	554
PRIMARY_HEAP_SIZE	554
QUIES_WAIT_TASK_TIME	555
QUIES_WAIT_USER_TIME	556
SEC_CLASS	556
SEC_RNAME	558
SEGMENT_POCKETS	559
SSB_CHAIN	560
TRANSPARENT_CHECKPOINT	560
WAIT_ALLOC_TIME	561
WINDOW_CHKPT_TIME	562
WINDOW_POCKETS	562
Chapter 9. Graphic Monitor Facility Host Subsystem Statements	565
API	565
CHECKPOINT	566
DOMAIN	567
GMTOFFSET	567
JAPANESE	568
LCON-AGG-BUNDLE-INTERVAL	568
LCON-AGGRST-REQUIRED	569
LCON-AIP-RESET-INTERVAL	569
LCON-ALERT-CMD-TIMEOUT	570
LCON-ASSOCIATE-NULL-NODE-WITH-LINK	571
LCON-EVCHANGE-BUFFER-INTERVAL	571
LCON-HEX-SUBVECTOR-DISPLAY	572
LCON-MAX-LOCATE-RESOURCE-VIEWS	572
LCON-MAX-QUEUE-DBSERVER	573
LCON-MAX-QUEUE-EVENTMGR	574
LCON-MAX-QUEUE-IPC	574
LCON-MAX-QUEUE-IRMGR	575
LCON-MAX-QUEUE-MAINTASK	576
LCON-MAX-QUEUE-NETCMD	577
LCON-MAX-QUEUE-NETCON	577
LCON-MAX-QUEUE-OPERIF	578
LCON-MAX-QUEUE-RCMGR	579

LCON-MAX-QUEUE-RTMGR	580
LCON-MAX-QUEUE-VIEWMGR	580
LCON-MAX-QUEUE-VSTATMGR	581
LCON-NCC-RETRY-LIMIT	582
LCON-NCC-RSC-LIMIT	583
LCON-NMG-POLL-INTERVAL	583
LCON-OPERATOR-CMD-AUDIT	584
LCON-REPORT-UNKNOWN-STATUS	585
LCON-SNATM-TIMEOUT	585
LCON-STATUS-DELAY-MAX	586
LCON-STATUS-DELAY-TIME	587
LEVEL	587
PRINTPDU38	588
RODMID	590
RODMNAME	590
STORAGE	591
TASK	591
TRACE	592
TRACEBYTES	593
TRACEPAGES	594
TYPE	594
Appendix. Common Global Variables for the Canzlog Archiving Function	597
Notices	603
Programming Interfaces	604
Trademarks	604
Privacy policy considerations	605
Index	607

About this publication

The IBM® Tivoli® NetView® for z/OS® product provides advanced capabilities that you can use to maintain the highest degree of availability of your complex, multi-platform, multi-vendor networks and systems from a single point of control. This publication, the *IBM Tivoli NetView for z/OS Administration Reference*, 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.

Intended audience

This publication is for operators and system programmers who need a complete understanding of the NetView program definition statements.

Publications

This section lists publications in the IBM Tivoli NetView for z/OS library and related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

IBM Tivoli NetView for z/OS library

The following documents are available in the IBM Tivoli NetView for z/OS library:

- *Administration Reference*, SC27-2869, describes the NetView program definition statements required for system administration.
- *Application Programmer's Guide*, SC27-2870, describes the NetView program-to-program interface (PPI) and how to use the NetView application programming interfaces (APIs).
- *Automation Guide*, SC27-2846, describes how to use automated operations to improve system and network efficiency and operator productivity.
- *Command Reference Volume 1 (A-N)*, SC27-2847, and *Command Reference Volume 2 (O-Z)*, SC27-2848, describe the NetView commands, which can be used for network and system operation and in command lists and command procedures.
- *Customization Guide*, SC27-2849, describes how to customize the NetView product and points to sources of related information.
- *Data Model Reference*, SC27-2850, provides information about the Graphic Monitor Facility host subsystem (GMFHS), SNA topology manager, and MultiSystem Manager data models.
- *Installation: Configuring Additional Components*, GC27-2851, describes how to configure NetView functions beyond the base functions.
- *Installation: Configuring Graphical Components*, GC27-2852, describes how to install and configure the NetView graphics components.
- *Installation: Configuring the NetView Enterprise Management Agent*, GC27-2853, describes how to install and configure the NetView for z/OS Enterprise Management Agent.
- *Installation: Getting Started*, GI11-9443, describes how to install and configure the base NetView program.

- *Installation: Migration Guide*, GC27-2854, describes the new functions that are provided by the current release of the NetView product and the migration of the base functions from a previous release.
- *IP Management*, SC27-2855, describes how to use the NetView product to manage IP networks.
- *Messages and Codes Volume 1 (AAU-DSI)*, GC27-2856, and *Messages and Codes Volume 2 (DUI-IHS)*, GC27-2857, describe the messages for the NetView product, the NetView abend codes, the sense codes that are included in NetView messages, and generic alert code points.
- *Programming: Assembler*, SC27-2858, describes how to write exit routines, command processors, and subtasks for the NetView product using assembler language.
- *Programming: Pipes*, SC27-2859, describes how to use the NetView pipelines to customize a NetView installation.
- *Programming: PL/I and C*, SC27-2860, describes how to write command processors and installation exit routines for the NetView product using PL/I or C.
- *Programming: REXX and the NetView Command List Language*, SC27-2861, describes how to write command lists for the NetView product using the Restructured Extended Executor language (REXX) or the NetView command list language.
- *Resource Object Data Manager and GMFHS Programmer's Guide*, SC27-2862, describes the NetView Resource Object Data Manager (RODM), including how to define your non-SNA network to RODM and use RODM for network automation and for application programming.
- *Security Reference*, SC27-2863, describes how to implement authorization checking for the NetView environment.
- *SNA Topology Manager Implementation Guide*, SC27-2864, describes planning for and implementing the NetView SNA topology manager, which can be used to manage subarea, Advanced Peer-to-Peer Networking, and TN3270 resources.
- *Troubleshooting Guide*, GC27-2865, provides information about documenting, diagnosing, and solving problems that occur in the NetView product.
- *Tuning Guide*, SC27-2874, provides tuning information to help achieve certain performance goals for the NetView product and the network environment.
- *User's Guide: Automated Operations Network*, SC27-2866, describes how to use the NetView Automated Operations Network (AON) component, which provides event-driven network automation, to improve system and network efficiency. It also describes how to tailor and extend the automated operations capabilities of the AON component.
- *User's Guide: NetView*, SC27-2867, describes how to use the NetView product to manage complex, multivendor networks and systems from a single point.
- *User's Guide: NetView Enterprise Management Agent*, SC27-2876, describes how to use the NetView Enterprise Management Agent.
- *User's Guide: NetView Management Console*, SC27-2868, provides information about the NetView management console interface of the NetView product.
- *Licensed Program Specifications*, GC31-8848, provides the license information for the NetView product.
- *Program Directory for IBM Tivoli NetView for z/OS US English*, GI11-9444, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.
- *Program Directory for IBM Tivoli NetView for z/OS Japanese*, GI11-9445, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS product.

- *Program Directory for IBM Tivoli NetView for z/OS Enterprise Management Agent*, GI11-9446, contains information about the material and procedures that are associated with installing the IBM Tivoli NetView for z/OS Enterprise Management Agent.

Related publications

You can find additional product information on the NetView for z/OS web site at <http://www.ibm.com/software/tivoli/products/netview-zos/>.

For information about the NetView Bridge function, see *Tivoli NetView for OS/390 Bridge Implementation*, SC31-8238-03 (available only in the V1R4 library).

Accessing terminology online

The IBM Terminology web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology web site at <http://www.ibm.com/software/globalization/terminology/>.

For NetView for z/OS terms and definitions, see the IBM Terminology web site. The following terms are used in this library:

NetView

For the following products:

- Tivoli NetView for z/OS version 6 release 2 modification 1
- Tivoli NetView for z/OS version 6 release 2
- Tivoli NetView for z/OS version 6 release 1
- Tivoli NetView for z/OS version 5 release 4
- Tivoli NetView for z/OS version 5 release 3
- Tivoli NetView for OS/390® version 1 release 4
- NetView releases that are no longer supported

CNMCMD

For the CNMCMRD member and the members that are included in it using the %INCLUDE statement

CNMSTYLE

For the CNMSTYLE member and the members that are included in it using the %INCLUDE statement

DSIOPF

For the DSIOPF member and the members that are included in it using the %INCLUDE statement

PARMLIB

For SYS1.PARMLIB and other data sets in the concatenation sequence

MVS™ For z/OS operating systems

MVS element

For the base control program (BCP) element of the z/OS operating system

VTAM®

For Communications Server - SNA Services

IBM Tivoli Network Manager

For either of these products:

- IBM Tivoli Network Manager
- IBM Tivoli OMNIbus and Network Manager

IBM Tivoli Netcool/OMNIbus

For either of these products:

- IBM Tivoli Netcool/OMNIbus
- IBM Tivoli OMNIbus and Network Manager

Unless otherwise indicated, topics to programs indicate the latest version and release of the programs. If only a version is indicated, the topic is to all releases within that version.

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

Using NetView for z/OS online help

The following types of NetView for z/OS mainframe online help are available, depending on your installation and configuration:

- General help and component information
- Command help
- Message help
- Sense code information
- Recommended actions

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli Documentation Central website at <https://www.ibm.com/developerworks/mydeveloperworks/wikis/home/wiki/Tivoli%20Documentation%20Central>

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File > Print** window that enables Adobe Reader to print letter-sized pages on your local paper.

Ordering publications

You can order many Tivoli publications online at <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>

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, contact your software account representative to order Tivoli publications. To locate the telephone number of your local representative, perform the following steps:

1. Go to <http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>.
2. Select your country from the list and click **Go**.
3. Click **About this site** to see an information page that includes the telephone number of your local representative.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. 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.

For additional information, see the Accessibility appendix in the *User's Guide: NetView*.

Service Management Connect

Connect, learn, and share with Service Management professionals: product support technical experts who provide their perspectives and expertise.

Access Service Management Connect at <http://www.ibm.com/developerworks/servicemanagement/z/>. Use Service Management Connect in the following ways:

- Become involved with transparent development, an ongoing, open engagement between other users and IBM developers of Tivoli products. You can access early designs, sprint demonstrations, product roadmaps, and prerelease code.
- Connect one-on-one with the experts to collaborate and network about Tivoli and the NetView community.
- Read blogs to benefit from the expertise and experience of others.
- Use wikis and forums to collaborate with the broader user community.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education website at <http://www.ibm.com/software/tivoli/education>.

Tivoli user groups

Tivoli user groups are independent, user-run membership organizations that provide Tivoli users with information to assist them in the implementation of Tivoli Software solutions. Through these groups, members can share information and learn from the knowledge and experience of other Tivoli users.

Access the Tivoli Users Group at <http://www.tivoli-ug.org>.

Downloads

Clients and agents, and several free NetView applications can be downloaded from the NetView for z/OS support web site:

<http://www.ibm.com/software/sysmgmt/products/support/IBMTivoliNetViewforzOS.html>

After you open the Support Portal page, perform the following steps:

1. Scroll down to the **Downloads** section and click the **view all** link.
2. On the Downloads for NetView for z/OS page, check the **Tool/Utility** box in the **Filter by topic** section on the left side.
3. Download the items based on your requirements.

These applications can help with the following tasks:

- Migrating customization parameters and initialization statements from earlier releases to the CNMSTUSR member and command definitions from earlier releases to the CNMCMDU member.
- Getting statistics for your automation table and merging the statistics with a listing of the automation table

- Displaying the status of a job entry subsystem (JES) job or canceling 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 Time Sharing Option (TSO) commands and receiving responses

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

Online

Access the Tivoli Software Support site at <http://www.ibm.com/software/sysmgmt/products/support/index.html?ibmprd=tivman>. Access the IBM Software Support site at <http://www.ibm.com/software/support/probsub.html>.

IBM Support Assistant

The IBM Support Assistant is a free local software serviceability workbench that helps you resolve questions and problems with IBM software products. The Support Assistant provides quick access to support-related information and serviceability tools for problem determination. To install the Support Assistant software, go to <http://www.ibm.com/software/support/isa/>.

Troubleshooting information

For more information about resolving problems with the NetView for z/OS product, see the *IBM Tivoli NetView for z/OS Troubleshooting Guide*.

Additional support for the NetView for z/OS product is available through the NetView user group on Yahoo at <http://groups.yahoo.com/group/NetView/>. This support is for NetView for z/OS customers only, and registration is required. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to obtain resolution.

Conventions used in this publication

This section describes the conventions that are used in this publication.

Revision codes

This publication uses the following revision codes, which are located in the left margins:

- | The pipe character | is used to indicate changes made for the December, 2014 modifications to the document.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes,

multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)

- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: “Use the word *that* to introduce a restrictive clause.”; letters as letters example: “The LUN address must start with the letter *L*.”)
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents...

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Operating system-dependent variables and paths

For workstation components, this publication uses the UNIX convention for specifying environment variables and for directory notation.

When using the Windows command line, replace `$variable` with `%variable%` for environment variables and replace each forward slash (/) with a backslash (\) in directory paths. The names of environment variables are not always the same in the Windows and UNIX environments. For example, `%TEMP%` in Windows environments is equivalent to `$TMPDIR` in UNIX environments.

Note: If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Syntax diagrams

The following syntax elements are shown in syntax diagrams. Read syntax diagrams from left-to-right, top-to-bottom, following the horizontal line (the main path).

- “Symbols”
- “Parameters” on page xxii
- “Punctuation and parentheses” on page xxii
- “Abbreviations” on page xxiii

For examples of syntax, see “Syntax examples” on page xxiii.

Symbols

The following symbols are used in syntax diagrams:

- Marks the beginning of the command syntax.
- Indicates that the command syntax is continued.

- | Marks the beginning and end of a fragment or part of the command syntax.
- Marks the end of the command syntax.

Parameters

The following types of parameters are used in syntax diagrams:

Required

Required parameters are shown on the main path.

Optional

Optional parameters are shown below the main path.

Default

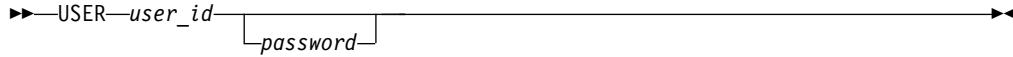
Default parameters are shown above the main path. In parameter descriptions, default parameters are underlined.

Syntax diagrams do not rely on highlighting, brackets, or braces. In syntax diagrams, the position of the elements relative to the main syntax line indicates whether an element is required, optional, or the default value.

When you issue a command, spaces are required between the parameters unless a different separator, such as a comma, is specified in the syntax.

Parameters are classified as keywords or variables. Keywords are shown in uppercase letters. Variables, which represent names or values that you supply, are shown in lowercase letters and are either italicized or, in NetView help, displayed in a differentiating color.

In the following example, the USER command is a keyword, the *user_id* parameter is a required variable, and the *password* parameter is an optional variable.



Punctuation and parentheses

You must include all punctuation that is shown in the syntax diagram, such as colons, semicolons, commas, minus signs, and both single and double quotation marks.

When an operand can have more than one value, the values are typically enclosed in parentheses and separated by commas. For a single value, the parentheses typically can be omitted. For more information, see “Multiple operands or values” on page xxiv.

If a command requires positional commas to separate keywords and variables, the commas are shown before the keywords or variables.

When examples of commands are shown, commas are also used to indicate the absence of a positional operand. For example, the second comma indicates that an optional operand is not being used:

`COMMAND_NAME opt_variable_1,,opt_variable_3`

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

Abbreviations

Command and keyword abbreviations are listed in synonym tables after each command description.

Syntax examples

The following examples show the different uses of syntax elements:

- “Required syntax elements”
- “Optional syntax elements”
- “Default keywords and values”
- “Multiple operands or values” on page xxiv
- “Syntax that is longer than one line” on page xxiv
- “Syntax fragments” on page xxiv

Required syntax elements:

Required keywords and variables are shown on the main syntax line. You must code required keywords and variables.

►—REQUIRED_KEYWORD—*required_variable*—►

A required choice (two or more items) is shown in a vertical stack on the main path. The items are shown in alphanumeric order.

►—[REQUIRED_OPERAND_OR_VALUE_1]
[REQUIRED_OPERAND_OR_VALUE_2]—►

Optional syntax elements:

Optional keywords and variables are shown below the main syntax line. You can choose not to code optional keywords and variables.

►—[OPTIONAL_OPERAND]—►

A required choice (two or more items) is shown in a vertical stack below the main path. The items are shown in alphanumeric order.

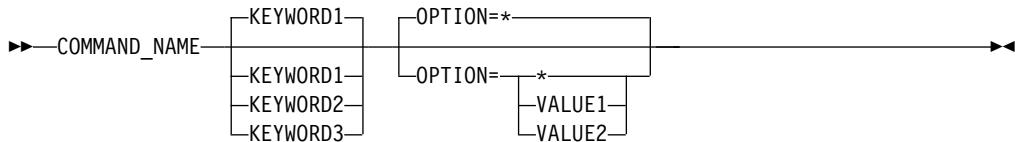
►—[OPTIONAL_OPERAND_OR_VALUE_1]
[OPTIONAL_OPERAND_OR_VALUE_2]—►

Default keywords and values:

Default keywords and values are shown above the main syntax line in one of the following ways:

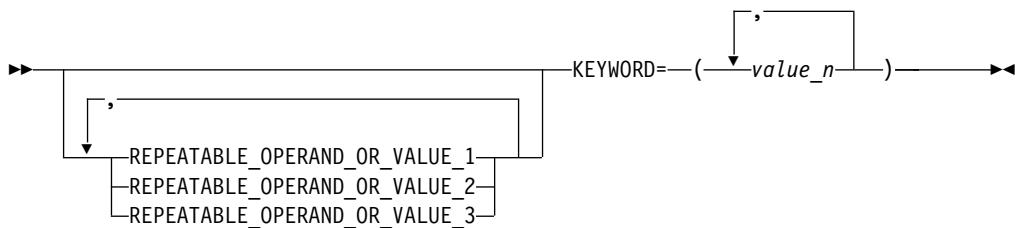
- A default keyword is shown only above the main syntax line. You can specify this keyword or allow it to default. The following syntax example shows the default keyword KEYWORD1 above the main syntax line and the rest of the optional keywords below the main syntax line.

- If an operand has a default value, the operand is shown both above and below the main syntax line. A value below the main syntax line indicates that if you specify the operand, you must also specify either the default value or another value shown. If you do not specify the operand, the default value above the main syntax line is used. The following syntax example shows the default values for operand OPTION=* above and below the main syntax line.



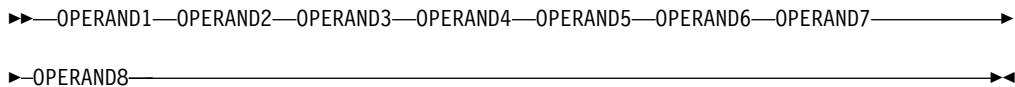
Multiple operands or values:

An arrow returning to the left above a group of operands or values indicates that more than one can be selected or that a single one can be repeated.



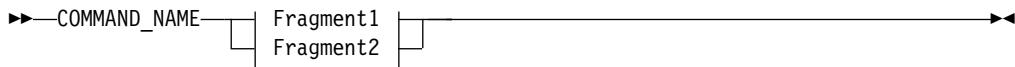
Syntax that is longer than one line:

If a diagram is longer than one line, each line that is to be continued ends with a single arrowhead and the following line begins with a single arrowhead.



Syntax fragments:

Some syntax diagrams contain syntax fragments, which are used for lengthy, complex, or repeated sections of syntax. Syntax fragments follow the main diagram. Each syntax fragment name is mixed case and is shown in the main diagram and in the heading of the fragment. The following syntax example shows a syntax diagram with two fragments that are identified as Fragment1 and Fragment2.



Fragment1

|—KEYWORD_A=valueA—KEYWORD_B—KEYWORD_C—|

Fragment2

|—KEYWORD_D—KEYWORD_E=*valueE*—KEYWORD_F—————|

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.

For information on changes to definition statements from previous releases, refer to the *IBM Tivoli NetView for z/OS Installation: Migration Guide*.

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 1 to find the sample name and its associated statements.

Table 1. Location of Statements within Samples and Location of Samples within the NetView Program

Sample	Statements	Page	Location
AAUCNMTD	DSTINIT	318	DSIPARM
AAUPRMLP	All statements are defined in the CNMSTYLE member and are referenced by the AAUPRMLP member.		DSIPARM
A01APPLS	STATOPT	391	VTAMLST
A01SNA	STATOPT	391	VTAMLST
BNJMBDST	All statements are defined in the CNMSTYLE member and are referenced by the BNJMBDST member.		DSIPARM
CNMCMD	%INCLUDE	333	DSIPARM
	CMDDEF	301	
	END	324	
CNMPOLCY	TCP390	476	DSIPARM
CNMSCNFT	%INCLUDE	333	DSIPARM
	ACTION	292	
	CMDLINE	308	
	COLUMNHEAD	310	
	HELD	327	
	HOLDPCNT	329	
	HOLDWARN	330	
	IMDAREA	331	
	INDENT	335	
	LASTLINE	341	
	LOCKIND	343	
	MLINDENT	349	
	NOPREFIX	367	

Table 1. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

Sample	Statements	Page	Location
	NORMAL	367	
	NORMQMAX	369	
	PREFIX	379	
	TITLE	396	
	TITLEDATE	398	
	TITLEDOMID	399	
	TITLEOPID	401	
	TITLESTAT	402	
	TITLETIME	404	
CNMSTYLE	See 7		DSIPARM
CNMSTPWD	PWD	207	DSIPARM
DSIALATD	ALIASMEM	293	DSIPARM
	DSTINIT	318	
DSIALTAB	COS	312	DSIPARM
	LU	345	
	MODE	350	
	ORIGNET	374	
DSIAMLTD	All statements are defined in the CNMSTYLE member and are referenced by the DSIAMLTD member.		DSIPARM
DSICNM	A (Alert)	289	DSIPARM
	C (Command List)	300	
	F (Filter)	324	
	M (Maximum)	346	
	O MONIT	370	
	O RESET	370	
	O SECSTAT	371	
	O SENDMSG	372	
	SENDMSG	388	
DSICPINT	DSTINIT	318	DSIPARM
	END	324	
DSICRTTD	DEFFOCPT	314	DSIPARM
	DSTINIT	318	
	END	324	
DSIDB2DF	SUBSYSTEM	395	DSIPARM
DSIELMEM	DSTINIT	318	DSIPARM
DSIELXIT	END	324	DSIPARM
DSIHINIT	DSTINIT	318	DSIPARM
	END	324	
	PARTNER	375	
	RETRY	386	

Table 1. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

Sample	Statements	Page	Location
DSIILGCF	All statements are defined in the CNMSTYLE member and are referenced by the DSIILGCF member.		DSIPARM
DSIKINIT	DSTINIT	318	DSIPARM
DSILOGBK	All statements are defined in the CNMSTYLE member and are referenced by the DSILOGBK member.		DSIPARM
DSILUCTD	All statements are defined in the CNMSTYLE member and are referenced by the DSILUCTD member.		DSIPARM
DSIOPF	%INCLUDE	333	DSIPARM
	END	324	
	OPERATOR	373	
	PROFILEN	383	
DSIPRFxx	AUTH	295	DSIPRF
	DOMAINS	317	
	END	324	
	ISPAN	336	
	PROFILE	382	
	SPAN	390	
DSIPROFxx	AUTH	295	DSIPRF
	DOMAINS	317	
	END	324	
	ISPAN	336	
	PROFILE	382	
	SPAN	390	
DSIQTSKI	CMDRCVR	309	DSIPARM
	REP	384	
	TASK	395	
DSIREXCF	All statements are defined in the CNMSTYLE member and are referenced by the DSIREXCF member.		DSIPARM
DSIRHOST	ACCESS	291	DSIPARM
DSIROVSI	DSTINIT	291	DSIPARM
DSIRSHCF	All statements are defined in the CNMSTYLE member and are referenced by the DSIRSHCF member.		DSIPARM
DSIRTTTD	All statements are defined in the CNMSTYLE member and are referenced by the DSIRTTTD member.		DSIPARM
DSISECUR	RMTSEC	386	DSIPARM
DSISVRTD	DSTINIT	318	DSIPARM
DSITCPCF	All statements are defined in the CNMSTYLE member and are referenced by the DSITCPCF member.		DSIPARM
DSITCPRF	NETCONV_IP	352	DSIPRF
	WEB_SERVER	407	
DSITRCBK	DSTINIT	318	DSIPARM
	LOGINIT	344	

Table 1. Location of Statements within Samples and Location of Samples within the NetView Program (continued)

Sample	Statements	Page	Location
DSIVPARM	DSTINIT	318	DSIPARM
	VPDINIT	406	
DSIWBMEM	All statements are defined in the CNMSTYLE member and are referenced by the DSIWBMEM member.		DSIPARM
DSI6INIT	DEFENTPT	313	DSIPARM
	DEFFOCPT	314	
	DSTINIT	318	
	RETRY	386	
DUIFPMEM	All statements are defined in the CNMSTYLE member and are referenced by the DUIFPMEM member.		DSIPARM
DUIGINIT	See 565		DSIPARM
DUIIGHB	All statements are defined in the CNMSTYLE member and are referenced by the DUIIGHB member.		DSIPARM
DUIISFP	AMELINIT	294	DSIPARM
	DSTINIT	318	
EKGCUST	See Chapter 8, "Resource Object Data Manager Definition Statements," on page 541		CNMSAMP
EZLCFG01	See Chapter 4, "Policy File Definitions," on page 409		DSIPARM
EZLINSMP	See Chapter 5, "Inform Policy Member," on page 493		DSIPARM

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:

- 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.
- In general, 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
```

One exception is with CNMSTYLE and CNMCMD definition statements, which specify a keyword whose value is a comma-delimited list. In this case, the value can be continued from one line to another by ending the line to be continued with a comma, and resuming the list value in column 2 of the next line, as shown in the following example:

```
ASSIGN.OPERGRP.GROUP = OPER1,OPER2,OPER3,OPER4,  
OPER5,OPER6
```

Another exception is with the webmenu statement; for information, see the usage notes in “webmenu” on page 276.

- 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) remains on your system definitions until you change those definitions and restart MVS.

Chapter 2. CNMSTYLE Initialization Statements

NetView initialization statements are defined in the CNMSTYLE member in the DSIPARM data set. To make changes to a CNMSTYLE statement, copy the statement to either the CNMSTUSR or CxxSTGEN member and then make any necessary updates. For more information on NetView initialization and updating CNMSTYLE statements, refer to *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Some CNMSTYLE statements are preceded with a tower parameter, or tower and subtower parameters. The syntax diagrams do not list the associated tower and subtower parameters. For example, the (TEMA.CONNACT) parameters are not listed with the syntax of the NACMD.INTCONNACT statement.

(TEMA.CONNACT)NACMD.INTCONNACT = 900

For tower and subtower information associated with a statement, review the CNMSTYLE member in the DSIPARM data set.

ACBpassword

Purpose

The ACBpassword statement specifies the password for ACBs for the NetView program if they have not previously been specified in an SAF product, such as the Resource Access Control Facility (RACF®), or in the NetView startup procedure, CNMSJ009. The password specified is used for the main application control block (ACB), for the primary program operator interface task (PPT), and for operator station tasks (OSTs).

Syntax

The ACBpassword statement has the following syntax:

ACBpassword

►—ACBpassword—=—*password*—►

where:

password

Specifies the password that is to be associated with the NetView ACBs. It can be up to eight characters in length and must match the value configured to VTAM in the APPL statement for NetView. It is set to the NetView domain name in the CNMSTPWD member provided by the NetView program.

Usage note

The ACBpassword statement must be included in member CNMSTPWD so that the password can be hidden. CNMSTPWD cannot be displayed by the BROWSE command.

AlertRcvName

Purpose

The AlertRcvName statement defines the PPI alert receiver name that is associated with the MOD=CNMCALRT task.

Syntax

The AlertRcvName statement has the following syntax:

AlertRcvName

►►—AlertRcvName—=—*name*————►►

where:

name

Specifies a 1 - 8 character PPI alert receiver name.

Usage notes

- 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 is set to the task name. For the task name CNMCALRT, the PPI name is always set to NETVALRT.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ALERTRCVNAME command.

ARCHIVE.ACCESSDELAY

Purpose

The ARCHIVE.ACCESSDELAY statement indicates the amount of time that the NetView program waits before attempting to access any archive data sets.

Syntax

The ARCHIVE.ACCESSDELAY statement has the following syntax:

ARCHIVE.ACCESSDELAY

►►—ARCHIVE.ACCESSDELAY = 00:00:00————►►

►►—ARCHIVE.ACCESSDELAY = *delay_time*————►►

where:

delay_time

Specifies the amount of time that the NetView program waits to access archive data sets. Specify the time in the format *hh:mm:ss*, where *hh* is the number of hours, *mm* is the number of minutes, and *ss* is the number of seconds.

The value must be between 00:00:00 and 23:59:59, inclusive. The default value is 00:00:00.

Usage notes

- The ARCHIVE.ACCESSDELAY statement only applies to NetView initialization. Use this statement when archive data sets cannot be successfully accessed until a system is in a state to allow access. For example, if the IBM Tivoli System Automation for z/OS product is used to start the Data Facility Hierarchical Storage Manager (DFSMShsm) program and one or more of the archive data sets was migrated, you must give enough time for the DFSMShsm program to start before the NetView program can access the archive data sets.
- During the delay time, archiving is not performed and users cannot access archived messages. When the time expires, a **RESTYLE ARCHIVE** command is issued.

When the **RESTYLE ARCHIVE** command is issued (either at the timer expiration or before), archive data sets can be accessed using the specified ARCHIVE.HLQ and ARCHIVE.WRITE values. If a **RESTYLE ARCHIVE** command is issued before the timer expires, the outstanding timer is canceled to avoid issuing another **RESTYLE ARCHIVE** command.

- If the ARCHIVE.ACCESSDELAY statement is omitted or the ARCHIVE.ACCESSDELAY statement has a value 00:00:00, archive data sets can be accessed as needed.
- To allow access to archive data sets as early as possible, implement automation to issue a **RESTYLE ARCHIVE** command as soon as access to data sets is allowed (for example, when the DFSMShsm program becomes active).

If you are using the System Automation for z/OS product, refer to its library for more information on issuing the **RESTYLE ARCHIVE** command. If you use automation, specify a value on ARCHIVE.ACCESSDELAY statement that is higher than the longest amount of time that it takes for automation to take place.

ARCHIVE.BROWSE.DATASPACES

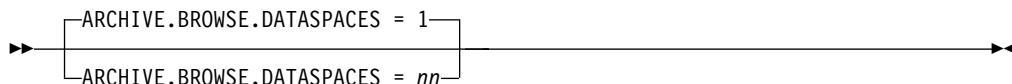
Purpose

The ARCHIVE.BROWSE.DATASPACES statement specifies the maximum number of data spaces that the NetView program can allocate for browsing the archived Canzlog data.

Syntax

The ARCHIVE.BROWSE.DATASPACES statement has the following syntax:

ARCHIVE.BROWSE.DATASPACES



where:

nn Specifies the maximum number of data spaces: 1 - 10, inclusive. The default value is 1.

Usage notes

- Ensure that at least 8M of space can be allocated for a data space or it cannot be used for browsing archived Canzlog data.

- To implement definition changes for the ARCHIVE.BROWSE.DATASPACES statement, the NetView program must be restarted.

ARCHIVE.HLQ

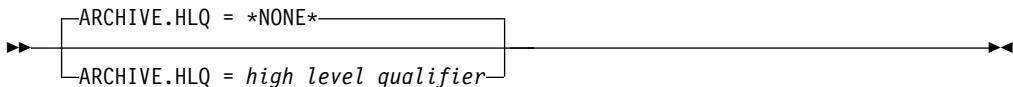
Purpose

The ARCHIVE.HLQ statement specifies the high-level qualifier for data sets that have or will have archived Canzlog data.

Syntax

The ARCHIVE.HLQ statement has the following syntax:

ARCHIVE.HLQ



where:

high_level_qualifier

Specifies the 1 - 26 character high-level qualifier. The high-level qualifier must conform to DFSMS data set naming conventions. The first character must be an alphabetic or a national character (@, #, \$). The value can include periods to separate qualifiers. The last character cannot be a period because all suffixes added by the NetView program begin with a period. Mixed case values are converted to upper case.

NONE

Specifies that this instance of the NetView program cannot write to the archive or read from it.

Usage notes

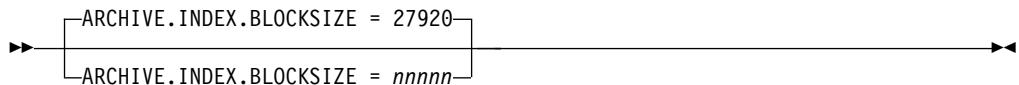
- The NetView program uses the high-level qualifier to begin the names of all the data sets in the archive. For information about the archive data set names, format, and contents, see *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.
- When you define a high-level qualifier, this instance of the NetView program can browse archived data, if there is data stored in the archive data set using the specified high-level qualifier.
- If you do not specify a high-level qualifier, then this instance of the NetView program cannot browse archived Canzlog data.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.HLQ statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.INDEX.BLOCKSIZE

The ARCHIVE.INDEX.BLOCKSIZE statement specifies the block size to use for the archive Canzlog primary index and index data sets.

The ARCHIVE.INDEX.BLOCKSIZE statement has the following syntax:

ARCHIVE.INDEX.BLOCKSIZE



where:

nnnnn

Specifies the blocksize: 80 - 32720, inclusive. Use a value that is a multiple of 80 (no remainder). The default value is 27920.

Usage notes

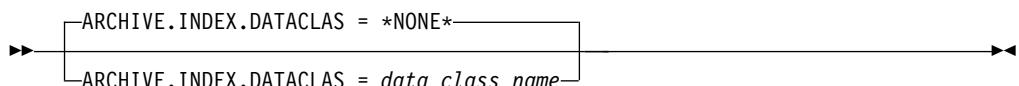
- If you are not using the SMS data class (or another selection method), use the ARCHIVE.INDEX.BLOCKSIZE statement to define a block size for a particular device type.
- Whether you specify a value for the ARCHIVE.INDEX.BLOCKSIZE statement or the default value is taken, the value is ignored if any of the following statements are coded with a value other than *NONE*:
 - ARCHIVE.INDEX.DATACLAS
 - ARCHIVE.INDEX.MGMTCLAS
 - ARCHIVE.INDEX.STORCLAS
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.INDEX.BLOCKSIZE statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.INDEX.DATACLAS

The ARCHIVE.INDEX.DATACLAS statement specifies the Storage Management Subsystem (SMS) data class for the Canzlog archive data sets that contain index data (individual index data sets and primary index data set).

The ARCHIVE.INDEX.DATACLAS statement has the following syntax:

ARCHIVE.INDEX.DATACLAS



where:

data_class_name

Specifies the 1 - 8 character SMS data class name

NONE

Specifies not to use a data class when the new index data set is allocated. This is the default value.

Usage notes

- The data class is used for the index data set attributes. The NetView program only validates the length of the data class name. The data class name is then validated by the operating system when the data set is allocated.
- If you specify a value other than *NONE* for the ARCHIVE.INDEX.DATACLAS, ARCHIVE.INDEX.MGMTCLAS, or ARCHIVE.INDEX.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive index data set characteristics and ignores the values of these statements:
 - ARCHIVE.INDEX.BLOCKSIZE
 - ARCHIVE.INDEX.SPACE
 - ARCHIVE.INDEX.UNIT
 - ARCHIVE.INDEX.VOLUME

Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.INDEX.DATACLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:

- Specify a logical record length (LRECL) 80.
- Specify a block size (BLKSIZE) that is an even multiple of the LRECL (80). Make sure that the block size is 80 - 32720, inclusive.
- Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
- Do not specify a retention period (RETPD) or expiration date (EXPDT).
- If you are not running or are not using the SMS facility to provide message data set characteristics, either omit the ARCHIVE.INDEX.DATACLAS statement or code a value of *NONE* to indicate that data class is not being used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.INDEX.DATACLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597

ARCHIVE.INDEX.MGMTCLAS

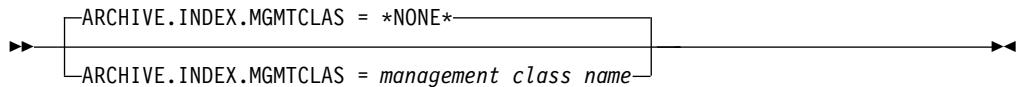
Purpose

The ARCHIVE.INDEX.MGMTCLAS statement specifies the Storage Management Subsystem (SMS) management class name for Canzlog archive data sets that contain index data (individual index data sets and primary index data set).

Syntax

The ARCHIVE.INDEX.MGMTCLAS statement has the following syntax:

ARCHIVE.INDEX.MGMTCLAS



where:

management_class_name

Specifies the 1 - 8 character name of the SMS management class name

NONE

Specifies not to use a management class when the new index data set is allocated. This is the default value.

Usage notes

- The management class is used for index data set management. The NetView program only validates the length of the management class name. The management class name is then validated by the operating system when the data set is allocated.
 - If you specify a value other than *NONE* for the ARCHIVE.INDEX.DATACLAS, ARCHIVE.INDEX.MGMTCLAS, or ARCHIVE.INDEX.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive index data set characteristics and ignores the values of these statements:
 - ARCHIVE.INDEX.BLOCKSIZE
 - ARCHIVE.INDEX.SPACE
 - ARCHIVE.INDEX.UNIT
 - ARCHIVE.INDEX.VOLUME
- Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.INDEX.MGMTCLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:
- Specify a logical record length (LRECL) 80.
 - Specify a block size (BLKSIZE) that is an even multiple of the LRECL (80). Make sure that the block size is 80 - 32720, inclusive.
 - Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
 - Do not specify a retention period (RETPD) or expiration date (EXPDT).
 - If you are not running or are not using the SMS facility to provide archive data set characteristics, either omit the ARCHIVE.INDEX.MGMTCLAS statement or code a value of *NONE* to indicate that data class is not being used.
 - To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
 - If you specify a valid value for the ARCHIVE.INDEX.MGMTCLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.INDEX.SPACE

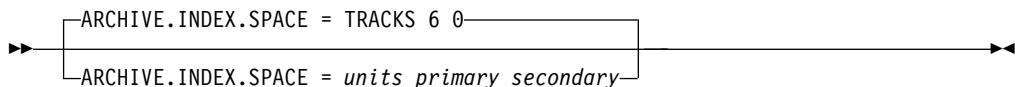
Purpose

The ARCHIVE.INDEX.SPACE statement specifies the space allocation parameters to use when the primary index and index data sets are allocated.

Syntax

The ARCHIVE.INDEX.SPACE statement has the following syntax:

ARCHIVE.INDEX.SPACE



where:

units

Specifies the units:

- BLOCKS (BLOCK, BLKS, BLK)
- CYLINDERS (CYLINDER, CYLS, CYL)
- TRACKS (TRACK, TRKS, TRK)

primary

Specifies the primary allocation. This is the number of space units (0 - 4096) that are used to allocate the primary data set extent for the primary index data set and each index data set.

secondary

Specifies the secondary allocation. This is the number of space units (0 - 4096) that are used to allocate the secondary data set extents for the primary index data set and each index data set. This parameter is optional. If it is not specified, the default value is 0.

Usage notes

- The primary and secondary allocation cannot both be zero (0).
- If you are not using the SMS data class (or another selection method), use the ARCHIVE.INDEX.SPACE statement to define the space allocation for the index data sets.
- Whether you specify a value for the ARCHIVE.INDEX.SPACE statement or the default value is taken, the value is ignored if any of the following statements are coded with a value other than *NONE*:
 - ARCHIVE.INDEX.DATACLAS
 - ARCHIVE.INDEX.MGMTCLAS
 - ARCHIVE.INDEX.STORCLAS
- If you do not specify the ARCHIVE.INDEX.SPACE statement and you do not provide SMS-related statements for the index data sets, then the NetView program uses the following default values:
TRACKS 6 0

For a 27920-byte block size (using a 3390 device type), an archive data set for Canzlog index data requires 6 tracks of space to hold the maximum number of 4096 80-byte records.

- If the NetView program detects an error with any individual parameter in the value specified for the ARCHIVE.INDEX.SPACE statement, the default value (TRACKS 6 0) is used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.INDEX.SPACE statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.INDEX.STORCLAS

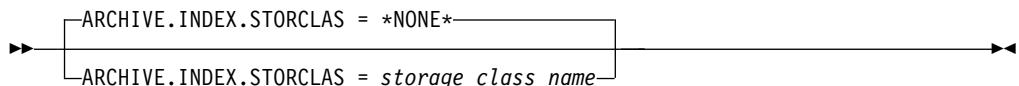
Purpose

The ARCHIVE.INDEX.STORCLAS statement specifies the Storage Management Subsystem (SMS) storage class name for the Canzlog archive index data sets.

Syntax

The ARCHIVE.INDEX.STORCLAS statement has the following syntax:

ARCHIVE.INDEX.STORCLAS



where:

storage_class_name

Specifies the 1 - 8 character name of the SMS storage class name

NONE

Specifies not to use a storage class when the new index data set is allocated

Usage notes

- The storage class is used for primary index data set or index data set storage attributes. The NetView program only validates the length of the storage class name. The storage class name is then validated by the operating system when the data set is allocated.
- If you specify a value other than *NONE* for the ARCHIVE.INDEX.DATACLAS, ARCHIVE.INDEX.MGMTCLAS, or ARCHIVE.INDEX.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive index data and primary index data set characteristics and ignores the values of these statements:
 - ARCHIVE.INDEX.BLOCKSIZE
 - ARCHIVE.INDEX.SPACE
 - ARCHIVE.INDEX.UNIT
 - ARCHIVE.INDEX.VOLUME

Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.INDEX.STORCLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:

- Specify a logical record length (LRECL) 80.
- Specify a block size (BLKSIZE) that is an even multiple of the LRECL (80). Make sure that the block size is 80 - 32720, inclusive.
- Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
- Do not specify a retention period (RETPD) or expiration date (EXPDT).
- If you are not running or are not using the SMS facility to provide message data set characteristics, either omit the ARCHIVE.INDEX.STORCLAS statement or code a value of *NONE* to indicate that storage class is not being used.
- If you do not specify the ARCHIVE.INDEX.STORCLAS statement or specify *NONE*, then the storage class is not specified when the new primary index data set or index data set is allocated.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.INDEX.STORCLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.INDEX.UNIT

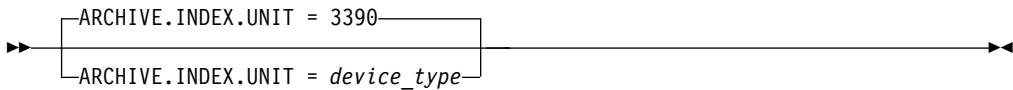
Purpose

The ARCHIVE.INDEX.UNIT statement specifies the unit name (device type) for the volume serial number specified on the ARCHIVE.INDEX.VOLUME statement. This can also be a name representing a group of volumes that the operating system uses when a new index data set allocation is requested. This statement is used for the allocation of Canzlog archive index data sets, including the primary index data set.

Syntax

The ARCHIVE.INDEX.UNIT statement has the following syntax:

ARCHIVE.INDEX.UNIT



where:

device_type

Specifies the 1 - 8 character unit name (device type). The default value is 3390.

The *device_type* can also be a name representing a group of volumes that the operating system uses when a new message data set allocation is requested.

Usage notes

- The unit name is validated by the operating system when the data sets are allocated.
- Whether you specify a value for the ARCHIVE.INDEX.UNIT statement or the default value is taken, the value is ignored if any of the following statements have a value other than *NONE*:
 - ARCHIVE.INDEX.DATACLAS
 - ARCHIVE.INDEX.MGMTCLAS
 - ARCHIVE.INDEX.STORCLAS
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.INDEX.UNIT statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see “Common Global Variables for the Canzlog Archiving Function,” on page 597.

ARCHIVE.INDEX.VOLUME

Purpose

The ARCHIVE.INDEX.VOLUME statement specifies a volume serial number to use when the index and primary index data sets are allocated.

Syntax

The ARCHIVE.INDEX.VOLUME statement has the following syntax:

ARCHIVE.INDEX.VOLUME

►►—ARCHIVE.INDEX.VOLUME = *volser*—►►

where:

volser

Specifies the 1- to 6-character volume serial number.

For more information about volume serial numbers, see *z/OS MVS JCL Reference*.

Usage notes

- The volume serial number is validated by the operating system when the data set is allocated.
- The value of the ARCHIVE.INDEX.VOLUME statement is ignored if any of the following statements have a value other than *NONE*:
 - ARCHIVE.INDEX.DATACLAS
 - ARCHIVE.INDEX.MGMTCLAS
 - ARCHIVE.INDEX.STORCLAS
- Because the ARCHIVE.INDEX.VOLUME statement is not required and does not have a default value, there must be another method (for example a storage class) that can be used to select a volume if the statement is not coded.

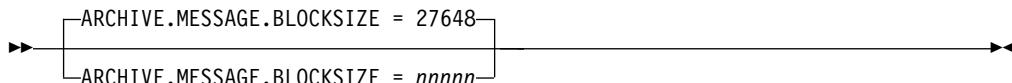
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE. INDEX. VOLUME statement, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see “Common Global Variables for the Canzlog Archiving Function,” on page 597.

ARCHIVE.MESSAGE.BLOCKSIZE

The ARCHIVE.MESSAGE.BLOCKSIZE statement specifies the block size to use when the Canzlog archive message data sets are allocated.

The ARCHIVE.MESSAGE.BLOCKSIZE statement has the following syntax:

ARCHIVE.MESSAGE.BLOCKSIZE



where:

nnnnn

Specifies the block size: 1024 - 31744, inclusive. Use a value that is a multiple of 1024 (no remainder). The default value is 27648.

Usage notes

- If you are not using the SMS data class (or another selection method), use the ARCHIVE.MESSAGE.BLOCKSIZE statement to define a block size for a particular device type.
- Whether you specify a value for the ARCHIVE.MESSAGE.BLOCKSIZE statement or the default value is taken, the value is ignored if any of the following statements are coded with a value other than *NONE*:
 - ARCHIVE.MESSAGE.DATACLAS
 - ARCHIVE.MESSAGE.MGMTCLAS
 - ARCHIVE.MESSAGE.STORCLAS
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.MESSAGE.BLOCKSIZE statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see “Common Global Variables for the Canzlog Archiving Function,” on page 597.

ARCHIVE.MESSAGE.DATACLAS

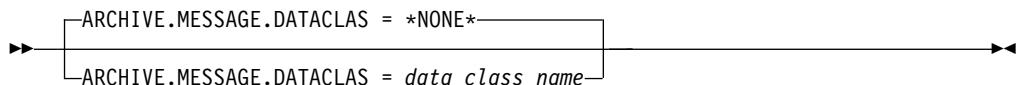
Purpose

The ARCHIVE.MESSAGE.DATACLAS statement specifies the Storage Management Subsystem (SMS) data class for the Canzlog archive message data sets.

Syntax

The ARCHIVE.MESSAGE.DATACLAS statement has the following syntax:

ARCHIVE.MESSAGE.DATACLAS



where:

data_class_name

Specifies the 1 - 8 character name of the SMS data class.

NONE

Specifies not to use a data class when a new message data set is allocated. This is the default value.

Usage notes

- The data class is used for the message data set attributes. The NetView program only validates the length of the data class name. The data class name is then validated by the operating system when the data set is allocated.
- If you specify a value other than *NONE* for the ARCHIVE.MESSAGE.DATACLAS, ARCHIVE.MESSAGE.MGMTCLAS, or ARCHIVE.MESSAGE.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive message data set characteristics and ignores the values of these statements:
 - ARCHIVE.MESSAGE.BLOCKSIZE
 - ARCHIVE.MESSAGE.SPACE
 - ARCHIVE.MESSAGE.UNIT
 - ARCHIVE.MESSAGE.VOLUMES

Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.MESSAGE.DATACLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:

- Specify a logical record length (LRECL) 1024.
- Specify a block size (BLKSIZE) that is an even multiple of the LRECL (1024). Make sure that the block size is 1024 - 31744, inclusive.
- Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
- Do not specify a retention period (RETPD) or expiration date (EXPDT).
- If you are not running or are not using the SMS facility to provide message data set characteristics, either omit the ARCHIVE.MESSAGE.DATACLAS statement or code a value of *NONE* to indicate that data class is not being used.
- If you do not specify the ARCHIVE.MESSAGE.DATACLAS statement or specify *NONE*, no data class is specified when the new message data set is allocated.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE

ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

- If you specify a valid value for the ARCHIVE.MESSAGE.DATACLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.MESSAGE.MGMTCLAS

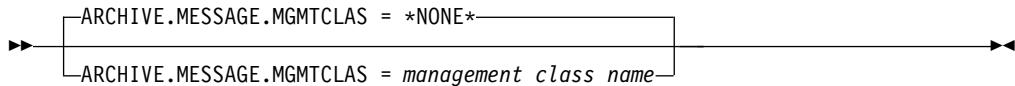
Purpose

The ARCHIVE.MESSAGE.MGMTCLAS statement specifies the Storage Management Subsystem (SMS) management class name for Canzlog archive message data sets.

Syntax

The ARCHIVE.MESSAGE.MGMTCLAS statement has the following syntax:

ARCHIVE.MESSAGE.MGMTCLAS



where:

management_class_name

Specifies the 1 - 8 character name of the SMS management class

NONE

Specifies not to use a management class when a new message data set is allocated. This is the default value.

Usage notes

- The management class is used for message data set management. The NetView program only validates the length of the management class name. The management class name is then validated by the operating system when the data set is allocated.
- If you specify a value other than *NONE* for the ARCHIVE.MESSAGE.DATACLAS, ARCHIVE.MESSAGE.MGMTCLAS, or ARCHIVE.MESSAGE.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive message data set characteristics and ignores the values of these statements:
 - ARCHIVE.MESSAGE.BLOCKSIZE
 - ARCHIVE.MESSAGE.SPACE
 - ARCHIVE.MESSAGE.UNIT
 - ARCHIVE.MESSAGE.VOLUMES

Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.MESSAGE.MGMTCLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:

- Specify a logical record length (LRECL) 1024.

- Specify a block size (BLKSIZE) that is an even multiple of the LRECL (1024). Make sure that the block size is 1024 - 31744, inclusive.
- Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
- Do not specify a retention period (RETPD) or expiration date (EXPDT).
- If you are not running or are not using the SMS facility to provide message data set characteristics, either omit the ARCHIVE.MESSAGE.MGMTCLAS statement or code a value of *NONE* to indicate that management class is not being used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.MESSAGE.MGMTCLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see “Common Global Variables for the Canzlog Archiving Function,” on page 597.

ARCHIVE.MESSAGE.SPACE

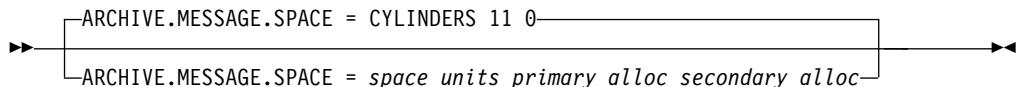
Purpose

The ARCHIVE.MESSAGE.SPACE statement specifies the space units and allocation parameters for the Canzlog archive message data sets.

Syntax

The ARCHIVE.MESSAGE.SPACE statement has the following syntax:

ARCHIVE.MESSAGE.SPACE



where:

space_units

Specifies the units:

- BLOCKS (BLOCK, BLKS, BLK)
- CYLINDERS (CYLINDER, CYLS, CYL)
- TRACKS (TRACK, TRKS, TRK)

primary_alloc

Specifies the number of space units (0-8192) to allocate for the primary extent of a message data set.

secondary_alloc

Specifies the secondary allocation. This is the number of space units (0 - 65536) that are used to allocate the secondary message data set extents. This parameter is optional. If it is not specified, the default value is 0.

Usage notes

- The primary and secondary allocation cannot both be zero (0).

- If you are not using the SMS data class (or another selection method), use the ARCHIVE.MESSAGE.SPACE statement to define the space allocation for the message data sets.
- Whether you specify a value for the ARCHIVE.MESSAGE.SPACE statement or the default value is taken, the value is ignored if any of the following statements are coded with a value other than *NONE*:
 - ARCHIVE.MESSAGE.DATACLAS
 - ARCHIVE.MESSAGE.MGMTCLAS
 - ARCHIVE.MESSAGE.STORCLAS
- If you do not specify the ARCHIVE.MESSAGE.SPACE statement and you do not provide SMS-related statements for the message data sets, then the NetView program uses the following default values:
`CYLINDERS 11 0`

For a 27648-byte block size (using a 3390 device type), an archive message data set requires 11 cylinders of space to hold the maximum number of 8192 1024-byte records.

- If the NetView program detects an error with any individual parameter in the value specified for the ARCHIVE.MESSAGE.SPACE statement, the default value (CYLINDERS 11 0) is used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.MESSAGE.SPACE statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see “Common Global Variables for the Canzlog Archiving Function,” on page 597.

ARCHIVE.MESSAGE.STORCLAS

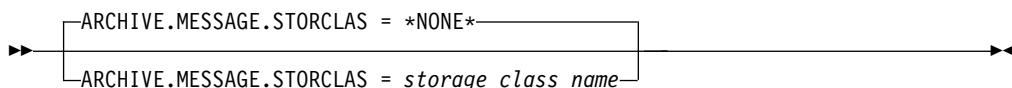
Purpose

The ARCHIVE.MESSAGE.STORCLAS statement specifies the Storage Management Subsystem (SMS) storage class name to be used for the Canzlog archive message data sets.

Syntax

The ARCHIVE.MESSAGE.STORCLAS statement has the following syntax:

ARCHIVE.MESSAGE.STORCLAS



where:

storage_class_name

Specifies the 1 - 8 character name of the SMS storage class

NONE

Specifies not to use a storage class when a new message data set is allocated. This is the default value.

Usage notes

- The storage class is used for the message data set storage attributes. The NetView program only validates the length of the storage class name. The storage class name is then validated by the operating system when the data set is allocated.
- If you specify a value other than *NONE* for the ARCHIVE.MESSAGE.DATACLAS, ARCHIVE.MESSAGE.MGMTCLAS, or ARCHIVE.MESSAGE.STORCLAS statement, the NetView program assumes that you are using SMS to provide archive message data set characteristics and ignores the values of these statements:
 - ARCHIVE.MESSAGE.BLOCKSIZE
 - ARCHIVE.MESSAGE.SPACE
 - ARCHIVE.MESSAGE.UNIT
 - ARCHIVE.MESSAGE.VOLUMES

Whether a data class is explicitly defined to the NetView program by using the ARCHIVE.MESSAGE.STORCLAS statement, a data class must be selected for data set allocation, and it must adhere to the following specifications:

- Specify a logical record length (LRECL) 1024.
- Specify a block size (BLKSIZE) that is an even multiple of the LRECL (1024). Make sure that the block size is 1024 - 31744, inclusive.
- Specify a space allocation that is large enough to accommodate the data set. Remember to consider the device type and block size when you determine the space allocation.
- Do not specify a retention period (RETPD) or expiration date (EXPDT).
- If you are not running or are not using the SMS facility to provide archive message data set characteristics, either omit the ARCHIVE.MESSAGE.STORCLAS statement or code a value of *NONE* to indicate that storage class is not being used.
- If you do not specify the ARCHIVE.MESSAGE.STORCLAS statement or specify *NONE*, no storage class is specified when the new message data set is allocated.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.MESSAGE.STORCLAS statement other than *NONE*, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.MESSAGE.UNIT

Purpose

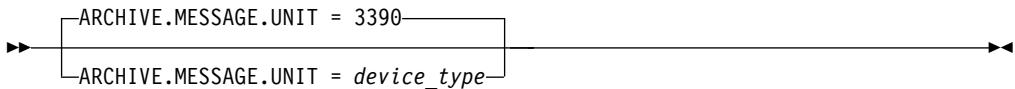
The ARCHIVE.MESSAGE.UNIT statement specifies the unit name (device type) for the volume serial number or numbers specified on the ARCHIVE.MESSAGES.VOLUMES statement. This can also be a name representing a group of volumes that the operating system uses when a new message data set

allocation is requested. This statement is used for the allocations of the Canzlog archive message data sets.

Syntax

The ARCHIVE.MESSAGE.UNIT statement has the following syntax:

ARCHIVE.MESSAGE.UNIT



where:

device_type

Specifies the 1 - 8 character unit name (device type). The default value is 3390.

The *device_type* can also be a name representing a group of volumes that the operating system uses when a new message data set allocation is requested.

Usage notes

- The unit name is validated by the operating system when the data sets are allocated.
- Whether you specify a value for the ARCHIVE.MESSAGE.UNIT statement or the default value is taken, the value is ignored if any of the following statements have a value other than *NONE*:
 - ARCHIVE.MESSAGE.DATACLAS
 - ARCHIVE.MESSAGE.MGMTCLAS
 - ARCHIVE.MESSAGE.STORCLAS
- If there are multiple volume serial numbers specified on the ARCHIVE.MESSAGE.VOLUMES statement, the unit name (device type) is used for all the volume serial numbers.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Whether you specify a valid value for the ARCHIVE.MESSAGE.UNIT statement or the default value is taken, the value is saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.MESSAGE.VOLUMES

Purpose

The ARCHIVE.MESSAGE.VOLUMES statement specifies 1 - 10 volume serial numbers to use when the Canzlog archive message data sets are allocated.

Syntax

The ARCHIVE.MESSAGE.VOLUMES statement has the following syntax:

ARCHIVE.MESSAGE.VOLUMES

```
>>ARCHIVE.MESSAGE.VOLUMES = [volser]
```

where:

volser

Specifies the 1- to 6-character volume serial number.

For more information about volume serial numbers, see *z/OS MVS JCL Reference*.

Usage notes

- The volume serial numbers are validated by the operating system when the data sets are allocated.
- The value of the ARCHIVE.MESSAGE.VOLUMES statement is ignored if any of the following statements have a value other than *NONE*:
 - ARCHIVE.MESSAGE.DATACLAS
 - ARCHIVE.MESSAGE.MGMTCLAS
 - ARCHIVE.MESSAGE.STORCLAS
- Because the ARCHIVE.MESSAGE.VOLUMES statement is not required and does not have a default value, there must be another method (for example a storage class) that can be used to select a volume if the statement is not coded.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If you specify a valid value for the ARCHIVE.MESSAGE.VOLUMES statement, it is saved in the value of a common global variable that may be examined by using the NetView program's services for accessing common global variables. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ARCHIVE.WRITE

Purpose

The ARCHIVE.WRITE statement specifies whether the NetView program is authorized to write to the Canzlog archive.

Syntax

The ARCHIVE.WRITE statement has the following syntax:

ARCHIVE.WRITE

```
>>[ARCHIVE.WRITE=Yes | ARCHIVE.WRITE>No]
```

where:

NO Specifies that this instance of the NetView program does not write to the Canzlog archive. Archived data can still be browsed.

YES

Specifies that this instance of the NetView program writes to the Canzlog archive. This is the default value.

Usage notes

- If the ARCHIVE.HLQ statement is omitted or has a value of *NONE*, then you must specify NO for the ARCHIVE.WRITE statement.
- Only one NetView program in an LPAR can write to the archive, even if multiple NetView programs are set up to write to the archive.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ARCHIVE command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- For the ARCHIVE.WRITE statement, a value will be saved in the value of a common global variable. You can use the NetView program's services for accessing common global variables to examine the value. For more information about the common global variables related to the Canzlog archiving function, see "Common Global Variables for the Canzlog Archiving Function," on page 597.

ASSIGN

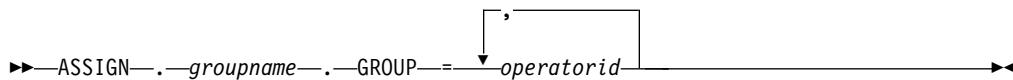
Purpose

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

Syntax

The ASSIGN statement has the following syntax:

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 notes

- You can assign up to a maximum of 255 operator IDs to each message receiver group.
- You can also use the ASSIGN command to assign operator IDs to groups. For more information, refer to the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help.

AUTOCMD

Purpose

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.

Syntax

The AUTOCMD statement has the following syntax:

AUTOCMD

```
►►AUTOCMD.—table_name.—►►  
          | ORDER=sort_value  
          | LIST=member_name  
          | MARKER=value
```

where:

table name

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 values:

FIRST

Inserts the table as the first automation table.

LAST

Inserts the table as the last automation table.

NONE

Causes the table not to be loaded.

You can use the *NONE* specification if you want to override an AUTOCMD.table_name.ORDER statement that is specified for the same table name in another CNMSTYLE member.

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. If you code multiple AUTOCMD entries with the LIST option, you must code unique values for LIST value.

MARKER

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

value Specifies the 1 - 8 character marker value. If the marker value is greater than 8 characters, it is truncated on the right to 8 characters.

If you code multiple AUTOCMD entries with the MARKER option, you must code unique values for MARKER value.

Usage notes

- If you add AUTOCMD statements that specify LIST or MARKER values, these values must be unique.
- If AUTOCMD.*table_name*.ORDER is omitted, the specified automation table is not loaded.
- AUTOCMD.*table_name*.MARKER is optional. The marker is used with the AUTOMAN command.
- You can use the AUTOTBL command to load and test automation tables. You can use the automation table management command (AUTOMAN) to work with individual or multiple automation tables through a full-screen panel interface. For more information, refer to the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

Important: If you modify the DSITBL01 automation table, consider that many of the statements are used in the normal operation of the NetView program.

AUTOTASK

Purpose

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

Syntax

The AUTOTASK statement has the following syntax:

AUTOTASK

```
►►—AUTOTASK—.—autotask_name—.—[Console=console_name]—[InitCmd=command]-►►
```

where:

autotask_name

Is the autotask name.

Console

Specifies the MVS console name that is to be associated with the autotask.

console_name

Specifies a 2 - 8 character console name. You can also specify the following values:

INTERN

You can use CONSOLE=*INTERN* to assign an autotask to respond to commands from the INTERNAL console.

ANY

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

MASTER

Use the *MASTER* value to assign an autotask to respond to commands from any console with master authority not otherwise assigned to an autotask. Use this value if you are not

using the *ANY* value or if you want consoles that have master authority to use a different autotask from consoles that have less authority.

NONE

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

InitCmd

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

command

Specifies the initial command.

Usage notes

- The autotask does not process commands until CNMSTYLE processing 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, see the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help.
 - You can use a question mark (?) followed by the function name instead of specifying an operator ID for the autotask, for example:
`AUTOTASK.?Primary.Console = *NONE*`

If you do this, use the `function.autotask` statement to define the operator ID:

```
function.autotask.primary = AUTO1
```

Note that if the name used with a question mark (Primary) is not specified in a function.autotask statement, the autotask statement is ignored. Function names can be a maximum length of 15 characters.

auxInitCmd

Purpose

The `auxInitCmd` statement defines a command or a command list to run automatically when the NetView program is started.

Syntax

The `auxInitCmd` statement has the following syntax:

►—auxInitCmd.—value—=—*command*
 └*NONE*┘

where

ue Specifies the EBCDIC value that determines the order in which the command

is to

mand

NONE

Overrides an auxInitCmd statement in an earlier section of the CNMSTYLE member.

Usage note

The commands specified on the auxInitCmd statements run before any commands for any autotask. All commands for autotasks, including both the task initial command list and commands sent by EXCMD, are queued and held up. They are run only after all auxInitCmd statements have completed. Messages are also queued; they are not submitted to automation or logged until all auxInitCmds have completed.

banner**Purpose**

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

Syntax

The banner statement has the following syntax:

banner

►—banner— —=— —*string*—————►

where:

string

Specifies up to 24 characters of data that is to 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 is not to 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 *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help.

CCDEF**Purpose**

The CCDEF statement specifies a DSIPARM member from which command definitions are to be read.

Syntax

The CCDEF statement has the following syntax:

CCDEF

►—CCDEF—=*member_name*—►

where:

member_name

Specifies the DSIPARM member name.

Usage note

For information about correlated command definitions, see sample DSICCDEF.

CNMI

Purpose

The CNMI statement defines whether this NetView program owns the Communications Network Interface (CNMI).

Syntax

The CNMI statement has the following syntax:

CNMI

►—

CNMI—=—Yes
CNMI—=—No

—►

where:

No Indicates that another NetView program has the CNMI. Specify No if you are configuring a secondary NetView program.

Yes

Indicates that this NetView program has the CNMI. This is the default.

Usage note

See *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information about configuring the multiple NetView programs.

COMMON.*variable_name*

Purpose

The COMMON.*variable_name* statement sets common global variables.

Syntax

The COMMON.*variable_name* statement has the following syntax:

COMMON.variable_name

►►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 *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help.

COMMON.CNMIP.DNSTimeout

Purpose

The COMMON.CNMIP.DNSTimeout statement specifies the maximum amount of time that a PING, TCPCONN, or TRACERTE command waits for an IP host name to be resolved.

Syntax

The COMMON.CNMIP.DNSTimeout statement has the following syntax:

COMMON.CNMIP.DNSTimeout

►►COMMON.—CNMIP.—DNSTimeout—=—seconds—►►

where:

seconds

A positive integer that specifies the number of seconds to wait. The initial value is 5 seconds.

Usage notes

To choose a value of the CNMIP.DNSTimeout statement that is sufficiently large to enhance the chances of resolver requests completing and returning results to their requestors, consider the following z/OS Communications Server settings:

- The number of domain name servers defined
- The resolver timeout value
- The number of times to try each name server
- The number of domain names to be searched

If the value of the IPv6Env CNMSTYLE statement is MIXED, a resolver request might take up to twice as long to complete as a request where only IPv4 or only

IPv6 is used. See the description of the RESOLVERUDPRETRIES statement in the *z/OS Communications Server: IP Configuration Reference, SC31-8776* for more information about calculating the maximum amount of time that can be used for resolver communication with the Domain Name Server (DNS) using the UDP protocol.

COMMON.CNMSNMP.MIBPATH

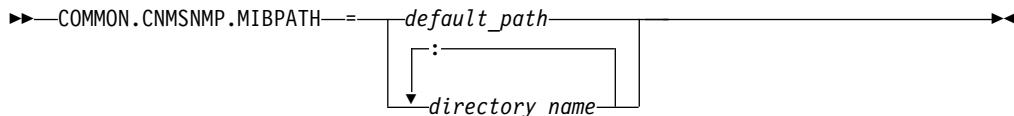
Purpose

The COMMON.CNMSNMP.MIBPATH statement specifies a list of UNIX system services file system directories where MIB source files are located. The MIB files in these directories are used by the SNMP command.

Syntax

The COMMON.CNMSNMP.MIBPATH statement has the following syntax:

COMMON.CNMSNMP.MIBPATH



where:

default_path

The list of directories where MIB source files are installed. The following path is the default path:

/usr/lpp/netview/v6r2m1/mibs:/etc/netview/mibs

If COMMON.CNMSNMP.MIBPATH is not specified, these directories are searched for any MIB files that need to be read by the SNMP command.

directory_name

The name of a directory in UNIX system services where MIB source files are found. Multiple path names can be specified by separating each path name with a colon. The *directory_name* entry cannot contain blanks.

Usage notes

- If a plus sign (+) symbol is the first character of the value specified, the name or names specified are prepended to the *default_path*. If a plus sign (+) is specified, it must be the first character. A blank cannot be specified between the plus sign and the directory it precedes.
- A colon (:) is a separator between directories. Do not use a blank before or after the colon.
- When an SNMP command is searching for a MIB, it uses the last directory in which the MIB was found.

Examples

1. The following specification results in /u/path01, /u/path02, and /u/path03 being searched when the SNMP command looks for MIB source files to read.

COMMON.CNMSNMP.MIBPATH=/u/path01:/u/path02:/u/path03

2. The following specification results in /u/path01, /u/path02, and /u/path03 being prepended to *default_path*. These directories are searched before *default_path* when the SNMP command reads MIB source files.

```
COMMON.CNMSNMP.MIBPATH=+/u/path01:/u/path02:/u/path03
```

Related statements

COMMON.CNMSNMP.MIBS

COMMON.CNMSNMP.MIBS

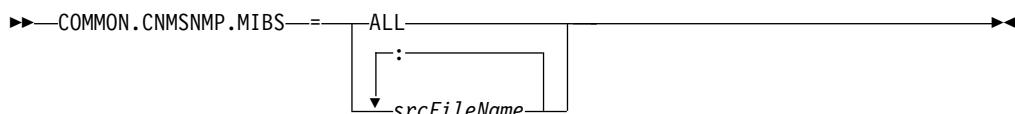
Purpose

The COMMON.CNMSNMP.MIBS statement specifies which MIB source files are to be read by the SNMP command.

Syntax

The COMMON.CNMSNMP.MIBS statement has the following syntax:

COMMON.CNMSNMP.MIBS



where:

ALL

Indicates that all MIB source files are to be read. ALL is the initial value that is specified in the COMMON.CNMSNMP.MIBS statement.

srcFileName

The name of a MIB source file. Multiple file names can be specified by separating each file name with a colon. The *srcFileName* cannot contain a blank character.

Usage note

A colon (:) is a separator between files names. Do not use a blank before or after the colon.

Examples

1. The following specification results in MIBS01, MIBS02, and MIBS03 being read by the SNMP command.

```
COMMON.CNMSNMP.MIBS=MIBS01:MIBS02:MIBS03
```

Related statements

COMMON.CNMSNMP.MIBPATH

COMMON.CNMSNMP.port

Purpose

The COMMON.CNMSNMP.port statement specifies the port on the SNMP agent for SNMP requests.

Syntax

The COMMON.CNMSNMP.port statement has the following syntax:

COMMON.CNMSNMP.port

►—COMMON—.—CNMSNMP—.—port—=—*port_number*————►

where:

port_number

The port number. The default is 161 for all requests except INFORM and TRAP, which default to 162.

COMMON.CNMSNMP.retries

Purpose

The COMMON.CNMSNMP.retries statement specifies the number of times that the NetView SNMP manager is to retry an SNMP request.

Syntax

The COMMON.CNMSNMP.retries statement has the following syntax:

COMMON.CNMSNMP.retries

►—COMMON—.—CNMSNMP—.—retries—=—*numRetry*————►

where:

numRetry

A positive integer indicating the number of times to retry SNMP commands. The initial value is 5 retries.

COMMON.CNMSNMP.timeout

Purpose

The COMMON.CNMSNMP.timeout statement specifies the maximum amount of time that the NetView SNMP manager is to wait for an SNMP agent to respond to a request.

Syntax

The COMMON.CNMSNMP.timeout statement has the following syntax:

COMMON.CNMSNMP.timeout

```
►►COMMON.—CNMSNMP.—timeout==seconds►►
```

where:

seconds

A positive integer that specifies the number of seconds to wait. The initial value is 1 second.

COMMON.CNMSNMP.trap

Purpose

The COMMON.CNMSNMP.trap statement specifies the port on the SNMP agent for SNMP TRAP requests

Syntax

The COMMON.CNMSNMP.trap statement has the following syntax:

COMMON.CNMSNMP.trap

```
►►[COMMON.—CNMSNMP.—trap==162]  
►►[COMMON.—CNMSNMP.—trap==port_number]►►
```

where:

port_number

The port number. The default value is 162.

COMMON.CNMTRAP.taskname.CONFIGFILE

Purpose

The COMMON.CNMTRAP.*taskname*.CONFIGFILE statement provides the name of a configuration file that contains SNMPv3 trap handling information. This configuration file can be an MVS data set or a UNIX System Services file. A configuration file is not required, but if a configuration file is not provided, then decryption and authentication of SNMPv3 traps are not done.

Syntax

The COMMON.CNMTRAP.*taskname*.CONFIGFILE statement has the following syntax:

COMMON.CNMTRAP.taskname.CONFIGFILE

```
►►COMMON.—CNMTRAP.—taskname.—CONFIGFILE==filename►►
```

where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated.

filename

Provides the name of an MVS data set or UNIX System Services file containing SNMPv3 trap-handling information. If the configuration file is a UNIX System Services file, specify the fully qualified path name. If an MVS data set name (either a sequential data set or a member of a partitioned data set) is to be specified, enclose the name in single quotation marks and precede it with slashes. If an MVS data definition (that is, a DD statement in the NetView start procedure or a file allocation using an ALLOCATE command) is to be specified, precede the file name with a DD prefix. These are some examples:

Note: The asterisk (*) in the first column serves to 'comment out' the statement. Remove the asterisk from any statement in order to activate that statement.

```
*COMMON.CNMTRAP.taskname.CONFIGFILE = //usr/lpp/netview/cnmtrapd.conf  
*COMMON.CNMTRAP.taskname.CONFIGFILE = //'USER.CNMTRAPD.CONF'  
*COMMON.CNMTRAP.taskname.CONFIGFILE = //'USER.CNMTRAPD.CONFPDS(CONFMEM)'  
*COMMON.CNMTRAP.taskname.CONFIGFILE = DD:CONFDD
```

Usage note

For more information about trap automation methodology, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.CNMTRAP.*taskname*.MAXTCPCONN

Purpose

The COMMON.CNMTRAP.*taskname*.MAXTCPCONN statement provides the maximum number of TCP connections supported by the SNMP trap automation task.

Syntax

The COMMON.CNMTRAP.*taskname*.MAXTCPCONN statement has the following syntax:

COMMON.CNMTRAP.*taskname*.MAXTCPCONN

```
►—COMMON—.—CNMTRAP—.—taskname—.—MAXTCPCONN—=—nnn—►►
```

where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated. If the SNMP trap automation task specified by *taskname* has not been enabled for receiving traps over TCP, the common global variable defined by this statement is ignored.

nnn

The maximum number of connections that an SNMP trap automation task supporting TCP can accept. The value range is 50 - 2000. If SNMP traps are to be received over TCP by this SNMP trap automation task and no value is provided for this variable, then a default value of 50 is used.

Usage note

For more information about this trap automation methodology, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.CNMTRAP.*taskname*.STACKNAME

Purpose

The COMMON.CNMTRAP.*taskname*.STACKNAME statement provides the name of the TCP/IP stack to which the SNMP trap automation task obtains affinity. If a name is not provided, then TCP/IP determines the actual stack affinity.

Syntax

The COMMON.CNMTRAP.*taskname*.STACKNAME statement has the following syntax:

COMMON.CNMTRAP.*taskname*.STACKNAME

►►—COMMON—.—CNMTRAP—.—*taskname*—.—STACKNAME—=—*tcpname*————►►

where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated.

tcpname

The name of a TCP/IP stack with which the SNMP trap automation task is to obtain affinity.

Usage note

For more information about this trap automation methodology, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.CNMTRAP.*taskname*.TCPPORT

Purpose

The COMMON.CNMTRAP.*taskname*.TCPPORT statement provides a port number for a concurrent server (TCP) to which clients can connect and send SNMP traps. If a port number is not provided, then SNMP traps are not received over TCP for this SNMP trap automation task.

Syntax

The COMMON.CNMTRAP.*taskname*.TCPPORT statement has the following syntax:

COMMON.CNMTRAP.*taskname*.TCPPORT

►►—COMMON—.—CNMTRAP—.—*taskname*—.—TCPPORT—=—*portnumber*————►►

where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated.

portnumber

The port number to be used by an SNMP trap automation task for receiving traps over TCP. The value range is 0 - 65535. If you specify a value of 0, TCP/IP chooses the port number.

For more information about trap automation methodology, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.CNMTRAP.*taskname*.TRACE

Purpose

The COMMON.CNMTRAP.*taskname*.TRACE statement enables or disables SNMP trap automation task tracing. If enabled, this also determines the level of detail. Trace messages are written to the NetView log.

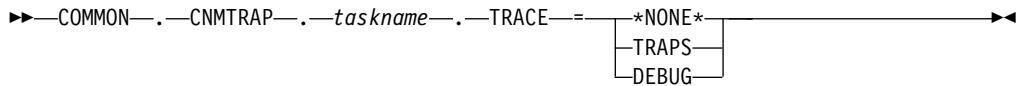
The trace is disabled for the following conditions:

- A value is not provided
- The value provided is not valid
- The value is *NONE*

Syntax

The COMMON.CNMTRAP.*taskname*.TRACE statement has the following syntax:

COMMON.CNMTRAP.*taskname*.TRACE



where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated.

NONE

The SNMP automation task does not write trace information to the NetView log.

TRAPS

Specifying TRAPS causes these results:

- The data is logged as it is received.
- When a complete protocol data unit (PDU) is collected, it is logged.
- If the data represents a valid SNMP trap, then the SNMP trap automation CP-MSU built from the trap is logged.

DEBUG

Specifying DEBUG is the equivalent of specifying TRAPS, and also provides additional detail of flows through the code.

Usage notes

- You must restart the SNMP trap automation task when you change the trace option.
- SNMP trap automation task tracing has no relationship to the tracing provided by the NetView TRACE command.
- For more information about this trap automation, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.CNMTRAP.*taskname*.UDPPORT

The COMMON.CNMTRAP.*taskname*.UDPPORT statement provides a port number to which entities can send SNMP trap datagrams (UDP). If a port number is not provided, then SNMP traps are not received over UDP for this SNMP trap automation task.

Note: If neither a TCP port (described in “COMMON.CNMTRAP.*taskname*.TCPPORT” on page 38) nor a UDP port definition is provided, then the SNMP trap automation task assumes that this is a UDP port and uses a default UDP port number of 162.

The COMMON.CNMTRAP.*taskname*.UDPPORT statement has the following syntax:

COMMON.CNMTRAP.*taskname*.UDPPORT

►—COMMON—.—CNMTRAP—.—*taskname*—.—UDPPORT—=—*portnumber*————►

where:

taskname

The name of the SNMP trap automation task with which the common global variable definition is associated.

portnumber

The port number to be used by an SNMP trap automation task for receiving traps over UDP. The value range is 0 - 65535. If you specify a value of 0, TCP/IP chooses the port number.

If this variable is not defined, then traps are not received over UDP by the SNMP trap automation task; if neither a TCPPORT (see “COMMON.CNMTRAP.*taskname*.TCPPORT” on page 38) or UDPPORT definition is provided, then the SNMP trap automation task assumes UDP-only and uses a default UDP port of 162.

For more information about this trap automation methodology, see the *IBM Tivoli NetView for z/OS Automation Guide*.

COMMON.DUIFHNAME

The COMMON.DUIFHNAME statement specifies the identifier for the GMFHS procedure. When you start a procedure in MVS, you can provide an identifier following the member name. For example, if you specify

S CNMGMFSH.GMFHS

then CNMGMFHS is the member name and GMFHS is the identifier. The identifier can be used to reference the task that is started in subsequent MVS commands. In some NetView documentation this might be referred to as an alias or a nickname.

The COMMON.DUIFHNAME statement has the following syntax:

COMMON.DUIFHNAME

►►—COMMON—.—DUIFHNAME—=—*identifier*—►►

where:

identifier

A sequence of 1 - 8 characters. The first character of this sequence must be alphabetic. The initial value for *identifier* is GMFHS.

Usage Note:

- If you do not use an identifier when GMFHS is started, comment out this statement.

COMMON.DUIFPRC

The COMMON.DUIFPRC statement specifies the name of a member that contains the source JCL for the GMFHS procedure. This information is used by command list CNME2101.

The COMMON.DUIFPRC statement has the following syntax:

COMMON.DUIFPRC

►►—COMMON.DUIFPRC—=—*member_name*—►►

where:

member_name

A sequence of 1 - 8 characters. The initial value for *member_name* is CNMGMFHS.

COMMON.EKGHNAM

The COMMON.EKGHNAM statement specifies the identifier for the RODM procedure. An identifier can be specified following the member name whenever an MVS procedure is started. For example, if you specify

S EKGXRODM.RODM

then EKGXRODM is the member name and RODM is the identifier. The identifier can be used to reference the procedure that is started in subsequent MVS commands. In some NetView documentation this might be referred to as an alias or a nickname.

The COMMON.EKGHNAM statement has the following syntax:

COMMON.EKGNAM

►►COMMON.—EKGNAM—=*identifier*►►

where:

identifier

A sequence of from 1 - 8 characters. The first character of the sequence must be alphabetic. The initial value for *identifier* is RODM.

COMMON.EKGHPRC

The COMMON.EKGHPRC statement specifies the name of a member that contains the source JCL for the RODM procedure. This information is used by command list CNME1098.

The COMMON.EKGHPRC statement has the following syntax:

COMMON.EKGHPRC

►►COMMON.—EKGHPRC—=*member_name*►►

where:

member_name

A sequence of 1 - 8 characters. The initial value for *member_name* is EKGXRODM.

COMMON.EZLINITDELAY

The COMMON.EZLINITDELAY statement defines the number of minutes that AON waits before starting to initialize.

The COMMON.EZLINITDELAY statement has the following syntax:

COMMON.EZLINITDELAY

►►COMMON.EZLINITDELAY—=*minutes*►►

where:

minutes

A positive integer that specifies the number of minutes to wait. If the value is 0 or if no value is specified, AON initializes without any delay.

COMMON.EZLIPTraceJCLWait

The COMMON.EZLIPTraceJCLWait statement specifies the amount of time that AON is to wait for responses to JCL errors. This statement is used by AON TCP CTRACE and packet trace external writer control.

The COMMON.EZLIPTraceJCLWait statement has the following syntax:

COMMON.EZLIPTraceJCLWait

►►—COMMON.EZLIPTraceJCLWait—=—seconds————►►

where:

seconds

A positive integer that specifies the number of seconds to wait. The initial value is 2 seconds.

COMMON.EZLRMTIMER

The COMMON.EZLRMTIMER statement specifies the interface to be used to access timers on remote domains.

The COMMON.EZLRMTIMER statement has the following syntax:

COMMON.EZLRMTIMER

►►—COMMON.—EZLRMTIMER—=—NETV—
—COMMON.—EZLRMTIMER—=—SA————►►

where:

NETV

Indicates that the NetView RMTCMD command is to be used to access remote timers. This is the default value if the COMMON.EZLRMTIMER statement is not specified. It is also the initial value for the COMMON.EZLRMTIMER statement.

SA Indicates that the SA/390 interface is to be used to access the remote timers.

COMMON.EZLsmtpHOSTNAME

Purpose

The COMMON.EZLsmtpHOSTNAME statement specifies the TCP/IP host name that is to be used in the FROM: and MAILFROM: header lines in email messages sent to the SMTP server.

Syntax

The COMMON.EZLsmtpNAME statement has the following syntax:

COMMON.EZLsmtpHOSTNAME

►►—COMMON.EZLsmtpHOSTNAME—=—tcpip_hostname————►►

where:

tcpip_hostname

A dotted name or IP address that identifies the TCP/IP host from which email is being sent.

If the COMMON.EZLsmtpHOSTNAME statement is specified, the value is used as the host name on the FROM: and MAIL FROM: header lines. If it is not specified, the email program (EZLEMAIL or EZLESMTTP) retrieves the value using SOCKET commands.

Usage notes

Consider the following information if you do not specify the COMMON.EZLsmtpHOSTNAME statement:

- The host name can only be retrieved if the host uses an IPv4 address.
- There can be a delay (up to the number of seconds specified on COMMON.CNMIP.DNSTimeout statement) for each SMTP call.
- The SOCKET calls use the stack name specified on the TCPNAME statement.

If any of these considerations are a problem, specify the COMMON.EZLsmtpHOSTNAME statement.

COMMON.EZLsmtpNAME

The COMMON.EZLsmtpNAME statement specifies the member name of the external writer that is to write an email message to the SMTP server.

The COMMON.EZLsmtpNAME statement has the following syntax:

COMMON.EZLsmtpNAME

►►COMMON.EZLsmtpNAME—=*member_name*►►

where:

member_name

A sequence of 1 - 8 characters. The initial value of *member_name* is SMTP.

COMMON.EZLTCPcTRACEwriter

The COMMON.EZLTCPcTRACEwriter statement specifies the name of the member containing the JCL to create the external writer for the AON TCP CTRACE.

The COMMON.EZLTCPcTRACEwriter statement has the following syntax:

COMMON.EZLTCPcTRACEwriter

►►COMMON.EZLTCPcTRACEwriter—=*member_name*►►

where:

member_name

A sequence of 1 - 8 characters. The initial value for *member_name* is CTTCP.

COMMON.EZLTCPPTRACEwriter

The COMMON.EZLTCPPTRACEwriter statement specifies the name of the member containing the JCL to create the external writer for the AON TCP packet trace.

The COMMON.EZLTCPPTRACEwriter statement has the following syntax:

COMMON.EZLTCPPTRACEwriter

```
►—COMMON.EZLTCPPTRACEwriter—=—member_name—►
```

where:

member_name

A sequence of 1 - 8 characters. The initial value of *member_name* is PKTCP.

COMMON.EZLTRACED

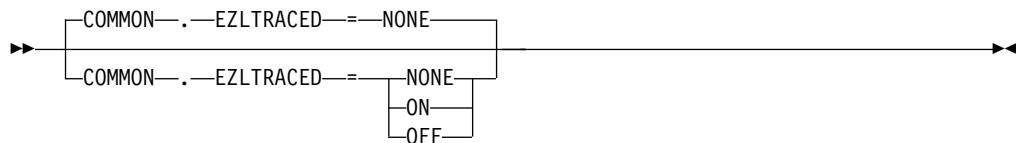
Purpose

The COMMON.EZLTRACED statement specifies the AON tracing options.

Syntax

The COMMON.EZLTRACED statement has the following syntax:

COMMON.EZLTRACED



where:

NONE

Disables tracing. Entry, exit, and program tracing are not performed. Specify this value for the best performance results. This is the default value.

ON Enables tracing and turns on entry/exit tracing. Program tracing can be turned on through the AON panels.

OFF

Enables tracing and turns off entry/exit tracing. Program tracing can be turned on through the AON panels.

Usage note

AON tracing can be overridden when the AON tower is enabled.

COMMON.FLC_DEF_NETW_VIEW

Purpose

The COMMON.FLC_DEF_NETW_VIEW statement specifies the default name and description of the network level view in which an object representing a network is to be displayed. The values specified for this variable are used if the NETWORK_VIEW keyword is not specified on a GETTOPO command. The MSM tower must be enabled. If this variable is not set, MSM defaults the values to MultiSysView/MultiSys LAN Networks.

Syntax

The COMMON.FLC_DEF_NETW_VIEW statement has the following syntax:

COMMON.FLC_DEF_NETW_VIEW

►—COMMON.FLC_DEF_NETW_VIEW—=—*viewName/desc*—►

where:

viewName

A sequence of characters up to 32 characters in length. The *viewname* specified can be up to 32 characters in length, and can be any alphanumeric character or any of these special characters:

@ \$. , () : ; ? ' - _ & + < >

The first character must be alphanumeric. This is the name that is displayed in the Network View list in the NetView management console.

desc

The *desc* value specified can be up to 32 characters in length, and can be any alphanumeric character or any of the special characters, except for a comma or an equal sign. It can include blank characters. This is the description displayed in the Description field of the NetView management console window.

COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS

Purpose

The COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS statement specifies the default status that is assigned to a real object that is created by the MultiSystem Manager for IBM Tivoli Network Manager component. This value is used as the displayStatus for real objects that do not have outstanding events. See the *IBM Tivoli NetView for z/OS Data Model Reference* for details about the displayStatus field. If you omit this statement, a value of 129 is used. The MSM tower must be enabled.

Syntax

The COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS statement has the following syntax:

COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS

►►—COMMON.FLC_DEFAULT_ITNM_OBJECT_STATUS—=—*status*—►►

where:

status

A number that indicates the default status. The following values are valid:

- 129 Satisfactory. Objects with this status are displayed as green in NetView management console views. This is the default value.
- 130 Unsatisfactory. Objects with this status are displayed as red in NetView management console views.
- 131 Intermediate. Objects with this status are displayed as yellow in NetView management console views.
- 132 Unknown Objects with this status are displayed as gray in NetView management console views.

COMMON.FLC_EXCEPTION_VIEW_FILE

Purpose

The COMMON.FLC_EXCEPTION_VIEW_FILE statement specifies the DSIPARM member name for the file containing the information necessary for the MultiSystem Manager to process exception views. The FLCSEXV member in the CNMSAMP data set provides sample exception view information. The MSM tower must be enabled.

Syntax

The COMMON.FLC_EXCEPTION_VIEW_FILE statement has the following syntax:

COMMON.FLC_EXCEPTION_VIEW_FILE

►►—COMMON.FLC_EXCEPTION_VIEW_FILE—=—*member_name*—►►

where:

member_name

A sequence of alphanumeric characters 1 - 8 characters in length.

COMMON.FLC_RODMAPPL

The COMMON.FLC_RODMAPPL statement specifies the user ID that is used to access RODM.

The COMMON.FLC_RODMAPPL statement has the following syntax:

COMMON.FLC_RODMAPPL

►►—COMMON.FLC_RODMAPPL—=—*applId*—►►

where:

appId

A sequence of characters that must match the ID defined to your SAF product. The value is case-sensitive. The default *appId* value is &DOMAIN.MSM.

Usage note

The COMMON.FLC_RODMNAME statement is used by the BLDVIEWS, Discovery Library Adapter (DLA), discovery manager, FLCARODM, Multisystem Manager, NetView management console, and NetView Resource Manager functions.

COMMON.FLC_RODMINT

The COMMON.FLC_RODMINT statement specifies the amount of time to wait between retries of a RODM request that has failed because RODM is checkpointing. The MSM tower must be enabled.

The COMMON.FLC_RODMINT statement has the following syntax:

COMMON.FLC_RODMINT

►►—COMMON.FLC_RODMINT—=—*seconds*—►►

where:

seconds

A positive integer between 0 and 15 that indicates the number of seconds to wait.

COMMON.FLC_RODMNAME

The COMMON.FLC_RODMNAME statement specifies the RODM name.

The COMMON.FLC_RODMNAME statement has the following syntax:

COMMON.FLC_RODMNAME

►►—COMMON.FLC_RODMNAME—=—*name*—►►

where:

name

A sequence of characters up to 8 characters in length. The name specified must be the same as the name assigned to the RODM task when it is started. The initial value of *name* is &CNMRODM.

Usage note

The COMMON.FLC_RODMNAME statement is used by the BLDVIEWS, Discovery Library Adapter (DLA), discovery manager, FLCARODM, Multisystem Manager, NetView Resource Manager, and the Visual BLDVIEWS functions. It is also used for NetView management console web launch when using %RODM.fieldname% substitution.

COMMON.FLC_RODMRETRY

The COMMON.FLC_RODMRETRY statement specifies the number of times to retry a RODM request that failed because RODM is checkpointing. The MSM tower must be enabled.

The COMMON.FLC_RODMRETRY statement has the following syntax:

COMMON.FLC_RODMRETRY

►►—COMMON.FLC_RODMRETRY—=—*numRetry*————►►

where:

numRetry

A positive integer between 0 and 5.

COMMON.FLC_RUNCMDRETRY

The COMMON.FLC_RUNCMDRETRY statement specifies the number of times a RUNCMD is retried after failing with a sense code of 0851 (indicating that the session is busy). The MSM tower must be enabled.

The COMMON.FLC_RUNCMDRETRY statement has the following syntax:

COMMON.FLC_RUNCMDRETRY

►►—COMMON.—FLC_RUNCMDRETRY—=—*numRetry*————►►

where:

numRetry

A positive integer between 0 and 10.

COMMON.FLC_TCPNAME

The COMMON.FLC_TCPNAME statement specifies the name of the TCP stack that is used for TCP/IP communications with service points. Code this statement if you are using TCP/IP over a connection that is not secure. If you omit this statement, the TCP/IP communications routine uses a default name of TCPIP. The MSM tower must be enabled.

The COMMON.FLC_TCPNAME statement has the following syntax:

COMMON.FLC_TCPNAME

►►—COMMON.—FLC_TCPNAME—=—*stackName*————►►

where:

stackName

A sequence of alphanumeric characters from 1 - 8 characters in length. The initial value of *stackName* is &CNMTCPN.

COMMON.IPPORTMON.INTVL

Purpose

The COMMON.IPPORTMON.INTVL statement specifies how frequently the IP port is monitored by the TESTPORT command.

Syntax

The COMMON.IPPORTMON.INTVL statement has the following syntax:

COMMON.IPPORTMON.INTVL

►►—COMMON.IPPORTMON.INTVL.*x*—=—*hh:mm:ss*—►►

where:

- *x* This is a positive integer and constitutes a numeric label to distinguish definitions. It must start with 1; consecutive numbers must be used.

hh:mm:ss

A valid timer interval that defines the frequency of the monitoring of the IP port.

Usage note

If you specify the COMMON.IPPORTMON.INTVL statement, you must specify the following statements:

- COMMON.IPPORTMON.IPADD
- COMMON.IPPORTMON.PORTNUM

Use the same numeric label (*x*) to link the COMMON.IPPORTMON statements together.

COMMON.IPPORTMON.IPADD

The COMMON.IPPORTMON.IPADD statement specifies the IP address associated with the IP port that is monitored by the TESTPORT command.

The COMMON.IPPORTMON.IPADD statement has the following syntax:

COMMON.IPPORTMON.IPADD

►►—COMMON.IPPORTMON.IPADD.*x*—=—*ip_address*—►►

where:

- *x* This is a positive integer and constitutes a numeric label to distinguish definitions. It must start with 1; consecutive numbers must be used.

ip_address

The IP address that is monitored by the TESTPORT command. It can be in any format, including IPv6 format.

Usage Note: If you specify the COMMON.IPPORTMON.IPADD statement, you must also specify the following statements:

- COMMON.IPPORTMON.INTVL
- COMMON.IPPORTMON.PORTNUM

Use the same numeric label (*x*) to link the COMMON.IPPORTMON statements together.

COMMON.IPPORTMON.PORTNUM

The COMMON.IPPORTMON.PORTNUM statement specifies the IP port that is monitored by the TESTPORT command.

The COMMON.IPPORTMON.PORTNUM statement has the following syntax:

COMMON.IPPORTMON.PORTNUM

►►—COMMON.IPPORTMON.PORTNUM.*x*—=—*portnum*—►►

where:

- *x* This is a positive integer and constitutes a numeric label to distinguish definitions. It must start with 1; consecutive numbers must be used.

portnum

Specifies the IP port which is monitored by the TESTPORT command.

Usage Note: If you specify the COMMON.IPPORTMON.PORTNUM statement, you must also specify the following statements:

- COMMON.IPPORTMON.IPADD
- COMMON.IPPORTMON.INTVL

Use the same numeric label (*x*) to link the COMMON.IPPORTMON statements together.

COMMON.IPPORTMON.PORTTO

The COMMON.IPPORTMON.PORTTO statement specifies the timeout value after which the TESTPORT command stops waiting for a response from the IP port if a response was not yet received.

The COMMON.IPPORTMON.PORTTO statement has the following syntax:

COMMON.IPPORTMON.PORTTO

►►—COMMON.IPPORTMON.PORTTO.*x*—=—*timeout_value*—►►

where:

- *x* This is a positive integer and constitutes a numeric label to distinguish definitions. It must start with 1; consecutive numbers must be used.

timeout_value

Specifies the timeout value in seconds.

Usage Note: If you specify the COMMON.IPPORTMON.PORTTO statement, you must also specify the following statements:

- COMMON.IPPORTMON.INTVL
- COMMON.IPPORTMON.IPADD

- COMMON.IPPORTMON.PORTNUM

Use the same numeric label (x) to link the COMMON.IPPORTMON statements together.

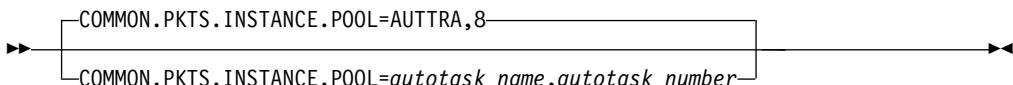
COMMON.PKTS.INSTANCE.POOL

The COMMON.PKTS.INSTANCE.POOL statement defines a pool of reusable autotasks for packet trace instances. This statement requires z/OS V2.1 or higher.

The COMMON.PKTS.INSTANCE.POOL statement is used by the IPTRACE and the PKTTRACE commands to dynamically allocate NetView autotasks to new packet trace instances when multiple packet trace instances are run on a stack. For additional information about running multiple packet trace instances, see *IBM Tivoli NetView for z/OS IP Management*.

The COMMON.PKTS.INSTANCE.POOL statement has the following syntax:

COMMON.PKTS.INSTANCE.POOL



where:

autotask_name

Specifies the base name for the autotasks. The default base name is AUTTRA.

autotask_number

Specifies the number of autotasks for concurrent packet trace instances. Each autotask can support only one packet trace instance at a time. Eight autotask definitions (AUTTRA1 through AUTTRA8) are supplied. The valid range is 1 - 32 autotasks.

Usage Notes:

- To use this statement, change the values in the CNMSTUSR or CxxSTGEN member. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- Multiple packet trace instances on a stack supports a maximum of 32 concurrent trace instances. To use more than eight autotasks, you must add more user definitions to the DSIOPFST member.

COMMON.SMFVPD

The COMMON.SMFVPD statement specifies an SMF record number used to log vital product data collected by the Network Asset Management commands (VPDPU, VPDDCE, VPDCMD) to an external log.

The COMMON.SMFVPD statement has the following syntax:

COMMON.SMFVPD



where:

smfRcrdId

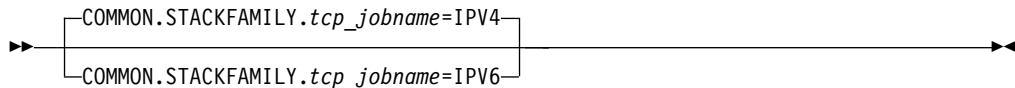
This is a positive integer that is a valid SMF record ID. To log the data collected by the Network Asset Management commands provided by NetView, specify a value of 37. If you customize these commands and want to use a different SMF record ID, you must use a value within the user-defined range of 128 - 255.

COMMON.STACKFAMILY

The COMMON.STACKFAMILY statement specifies the IP family to use for correlation methods in RODM and for displaying the IP address field in the NetView management console Resource Properties window. The IP family value is also displayed in the Stack Configuration and Status workspace in the NetView for z/OS Enterprise Management Agent and is included in the output of the STACSTAT command.

The COMMON.STACKFAMILY statement has the following syntax:

COMMON.STACKFAMILY



where:

IPV4

Specifies that the IP family for the specified TCP/IP stack is IPv4. This is the default setting.

IPV6

Specifies that the IP family for the specified TCP/IP stack is IPv6.

tcp_jobname

Specifies the job name of the TCP/IP stack for which IP family is to be set.

COMMON.WAITTIME

Purpose

The COMMON.WAITTIME statement specifies the amount of time that the AON component and System Automation for z/OS waits for responses from same-system commands.

Syntax

The COMMON.WAITTIME statement has the following syntax:

COMMON.WAITTIME

```
►► COMMON.WAITTIME==seconds ►►
```

where:

seconds

An integer that specifies the number of seconds to wait. The initial setting is 29 seconds.

COMMON.XDOMTIME

Purpose

The COMMON.XDOMTIME statement specifies the amount of time that the AON component and System Automation for z/OS waits for responses from cross-domain commands.

Syntax

The COMMON.XDOMTIME statement has the following syntax:

COMMON.XDOMTIME

►►—COMMON.XDOMTIME—=—*seconds*————►►

where:

seconds

An integer that specifies the number of seconds to wait. The initial setting is 210 seconds.

ConsFixed

Purpose

The ConsFixed statement can be used to specify a constant string that can be referenced by ConsMask to generate a console name.

Syntax

The ConsFixed statement has the following syntax:

ConsFixed

►►—ConsFixed—=—*string*————►►

where:

string

Specifies a sequence of characters that can be validly used in a console name.

Usage note

See “ConsMask” on page 55 for additional information about specifying a value for a console name.

ConsMask

Purpose

The value specified for ConsMask is used to create MVS extended console names. The combination of characters specified assists in assigning appropriate security to console names chosen by the NetView program and in finding a unique name for each console obtained.

NetView uses the value specified for ConsMask whenever it needs to generate a console name. These are some situations where NetView generates console names:

- An MVS command is issued by a task that has not obtained an MVS console using the SETCONID command.
- A GETCONID command is issued without the CONSOLE keyword specified and the task has not issued a prior SETCONID command.

NetView generates the console name by substituting the characters in the mask with characters from values for particular environment attributes, such as the current operator ID or the current domain ID. Each character in the mask is associated with a specific attribute in the environment. If the substitution results in a duplicate console name, NetView tries to resolve it by selecting a character to append at the end of the console name, if it is shorter than eight characters in length, or to overlay the eighth character of the console name, if it is eight characters in length. The selected character is one of the 36 alphanumeric characters or one of the three national characters: @, #, \$.

Syntax

The ConsMask statement has the following syntax:

ConsMask

►—ConsMask— —=— —mask—————►

where:

mask

Is an 8 - character sequence. The following characters are allowed:

- * A single * with no other characters. This selects a console name equal to the current operator ID.
- a Selects a character in the current operator ID. The first **a** in the mask selects the first character in the operator ID. Each subsequent **a** in the mask selects the next character in the operator ID in a forward direction.
- b Selects a character in the current operator ID. The first **b** in the mask selects the last character in the operator ID. Each subsequent **b** in the mask selects the next character in the operator ID in a backward direction.
- d Selects a character in the domain ID. The first **d** in the mask selects the first character in the domain ID. Each subsequent **d** in the mask selects the next character in the domain ID in a forward direction.
- e Selects a character in the domain ID. The first **e** in the mask selects the last character in the domain ID. Each subsequent **e** in the mask selects the next character in the domain ID in a backward direction.
- f Selects a character in the value specified for ConsFixed. The first **f** in

- the mask selects the first character specified for ConsFixed. Each subsequent **f** in the mask selects the next character specified for ConsFixed in a forward direction.
- m** Selects a character in the VTAM APPL ID for the current task. The first **m** in the mask selects the first character in the application ID. Each subsequent **m** in the mask selects the next character in the application ID in a forward direction.
 - n** Selects a character in the VTAM APPL ID for the current task. The first **n** in the mask selects the last character in the application ID. Each subsequent **n** in the mask selects the next character in the application ID in a backward direction.
 - 2** Selects a character in the value specified for the &NV2I variable. The first **2** in the mask selects the first character specified for &NV2I. Each subsequent **2** in the mask selects the next character specified for &NV2I in a forward direction.

Usage notes

- Except for the * character, you can use each character more than once. If the value for the environment information represented by one of the mask characters is shorter in length than the number of characters in the mask that are associated with that environment value, then the extra characters in the mask are ignored. For example, if the operator ID is **OP1**, the domain ID is **NV55**, and the mask has a value of **aaaaddddd**, then the generated console name is **OP1NV55**.
- The initial setting of the ConsMask statement is **aaaab22**.
- When a value other than the * character is specified, the value specified for CONSENTRY in any operator profile is overridden with the value that results from using the ConsMask statement.

Examples:

1. If the current operator ID is **OP01**, the domain ID is **NV55**, ConsFixed has a value of **MNLXYZ**, and NV2I has a value of **99**, then
 - A mask with a value of **aaaadddd** results in a console name of **OP01NV55**.
 - A mask with a value of **bbbbeeee** results in a console name of **10P055VN**.
 - A mask with a value of **adadadad** results in a console name of **ONPV0515**.
 - A mask with a value of **aaaab22** results in a console name of **OP01199**.
 - A mask with a value of **eefffaa22** results in a console name of **55MN0P99**.

CORRELATION

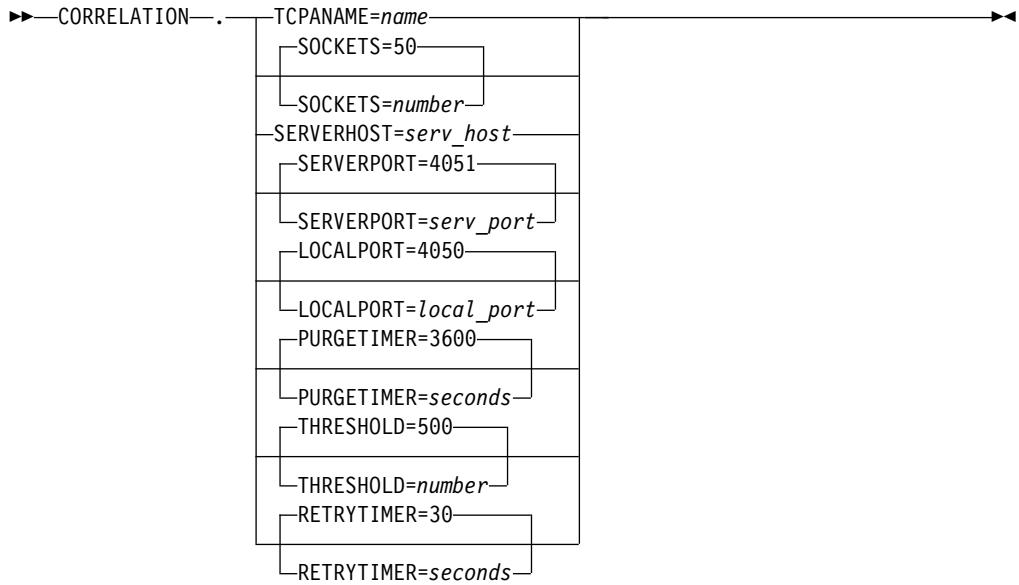
Purpose

The CORRELATION statement defines definitions for the communications task (DSICORSV) to the correlation engine running on UNIX System Services.

Syntax

The CORRELATION statement has the following syntax:

CORRELATION



where:

LOCALPORT=local_port

Defines the port number on which DSIREXCF waits for connection requests. The default is 4050.

PURGETIMER=seconds

Defines the time the NetView program is to keep a copy of the messages and MSUs that were sent to the correlation engine. The value is in seconds. The default is 3600 seconds.

RETRYTIMER=seconds

Defines the time that NetView keeps a message or MSU on a retry queue in the case of a failure sending the message or MSU to the correlation engine. The value is in seconds. The default is 30.

SERVERHOST=serv_host

Defines the host name that the correlation engine uses for TCP connections. This keyword has no default value.

SERVERPORT=serv_port

Defines the port number on which the correlation engine waits for connection requests from z/NetView. The default is 4051.

SOCKETS=number

Specifies the maximum number of simultaneous requests. The default is 50.

TCPNAME=name

Defines the TCP/IP address space name. The &CNMTCPN variable is the initial value specified in the CNMSTYLE member.

THRESHOLD=number

Defines the maximum number of messages and MSUs that the NetView program is to keep on the retry queue going to the correlation engine. The default is 500.

Usage notes

The LOCALPORT and SERVERPORT statements must be synchronized with the port properties in the FLBcorrelator.properties file in the correlation engine root

directory on the correlation engine running on UNIX System Services.

DB2SEC

Purpose

You can use the DB2SEC statement in the CNMSTYLE member to load the following interfaces:

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

After the CAF or RRS interfaces are loaded, tasks can access DB2 directly without needing the DSIDB2MT task.

Usage Note:

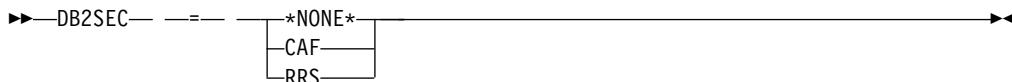
For tasks that access the DB2 subsystem that DSIDB2MT accesses, DSIDB2MT must still be active.

Starting the DSIDB2MT task ensures SQL requests that do not specify which DB2 to access, always run the same DB2. The SSID ssidname|* parameter on the SQL stage defines whether a specific or the default DB2 is accessed. When 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.

Syntax

The DB2SEC statement has the following syntax:

DB2SEC



where:

CAF

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

NONE

Specifies not to use DB2. This is the initial setting in the CNMSTYLE member.

RRS

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

DEFAULTS

Purpose

The DEFAULTS statement sets initial values for defaults used across all of the NetView program.

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.

Syntax

The DEFAULTS statement has the following syntax:

DEFAULTS

►—DEFAULTS—.—*keyword*— —=— —*value*—————►

where:

keyword

Specifies the DEFAULTS command keyword to be initialized. For more information about the keywords you can specify, refer to the DEFAULTS command in *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

value

Specifies the DEFAULTS command keyword value to be used. For more information about the values you can specify, refer to the DEFAULTS command in *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

Usage notes

- The HCYLOG, NETLOG, SYSLOG initial settings take effect after the corresponding logging task becomes active.
- Use the Banner statement instead of the DEFAULTS statement to initialize the banner.

DISCOVERY.INTAPPL

Purpose

The DISCOVERY.INTAPPL statement defines the time interval for data collectors to collect NetView application definition and status data.

Syntax

The DISCOVERY.INTAPPL statement has the following syntax:

DISCOVERY.INTAPPL

►—DISCOVERY.INTAPPL=300—————►

DISCOVERY.INTAPPL=*interval*

where:

interval

Specifies the time interval in the range 30 - 43200 seconds. The default value is 300 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DISCOVERY command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DISCOVERY.INTINTERFACE

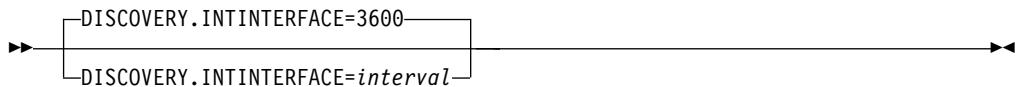
Purpose

The DISCOVERY.INTINTERFACE statement defines the time interval for data collectors to collect information about TCP/IP stack interfaces.

Syntax

The DISCOVERY.INTINTERFACE statement has the following syntax:

DISCOVERY.INTINTERFACE



where:

interval

Specifies the time interval in the range 300 - 43200 seconds. The default value is 3600 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DISCOVERY command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DISCOVERY.INTTELNET

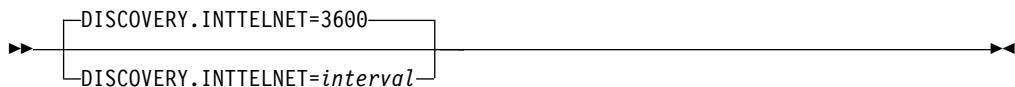
Purpose

The DISCOVERY.INTTELNET statement defines the time interval for data collectors to collect information about Telnet servers and Telnet ports.

Syntax

The DISCOVERY.INTTELNET statement has the following syntax:

DISCOVERY.INTTELNET



where:

interval

Specifies the time interval in the range 300 - 43200 seconds. The default value is 3600 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DISCOVERY command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DISCOVERY.NetViewOnly

Purpose

The DISCOVERY.NetViewOnly statement indicates whether to limit resource discovery to resources related to this instance of the NetView program. Use this statement to limit the number of resources being discovered when you have secondary NetView programs on the same z/OS image.

Syntax

The DISCOVERY.NetViewOnly statement has the following syntax:

DISCOVERY.NetViewOnly



where:

NO Specifies that all resources that are associated with the main DISCOVERY tower (not the subtowers) in the CNMSTYLE member are discovered. This is the default setting.

YES

Specifies to limit resource discovery to resources related to this instance of the NetView program. A small set of required system and sysplex resources are also discovered.

DISCOVERY.SNMP

Purpose

The DISCOVERY.SNMP statement indicates whether the discovery processes can attempt to issue SNMP commands to collect data. If this statement is not set to YES, some data will not be collected and will therefore be missing in the user interfaces.

Syntax

The DISCOVERY.SNMP statement has the following syntax:

DISCOVERY.SNMP



where:

NO Specifies not to collect the data. If you specify NO, data might be missing in user interfaces.

YES

Specifies to collect the data. This is the default setting.

DLA.cmdb_ftp_server

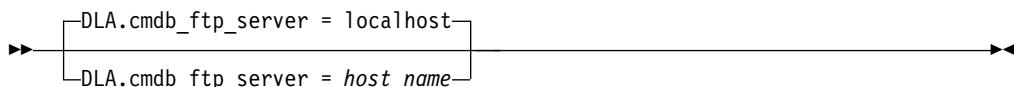
Purpose

The DLA.cmdb_ftp_server statement is used by the Discovery Library Adapter (DLA) to define the name of the Configuration Management Database (CMDB) FTP server.

Syntax

The DLA.cmdb_ftp_server statement has the following syntax:

DLA.cmdb_ftp_server



where:

host_name

Specifies the host name or IP address of the CMDB FTP server. The default value is localhost.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.debug

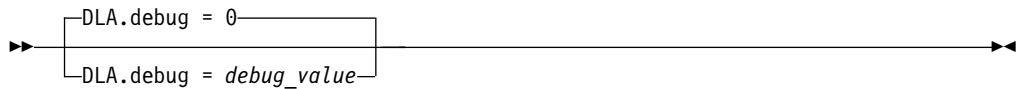
Purpose

The DLA.debug statement is used for debugging the Discovery Library Adapter (DLA).

Syntax

The DLA.debug statement has the following syntax:

DLA.debug



where:

debug_value

Specifies whether debugging is turned on (1) or off (0). The default value is 0.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_blocksize

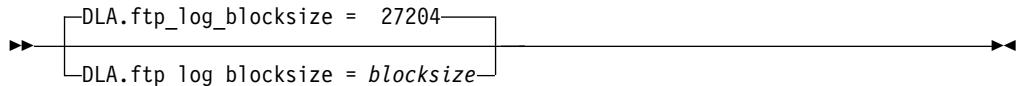
Purpose

The DLA.ftp_log_blocksize statement is used by the Discovery Library Adapter (DLA) during the allocation of the temporary FTP log file. This parameter specifies the block size and average record length for a data control block (DCB).

Syntax

The DLA.ftp_log_blocksize statement has the following syntax:

DLA.ftp_log_blocksize



where:

blocksize

The blocksize can contain a value from 0 - 65536. The default value is 27204.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_filename

Purpose

The DLA.ftp_log_filename statement is used by the Discovery Library Adapter (DLA) to define the data set in which to log output from the TSO FTP client.

Syntax

The DLA.ftp_log_filename statement has the following syntax:

DLA.ftp_log_filename

```
►►DLA.ftp_log_filename = ftp_log_file_name►►
```

where:

ftp_log_file_name

Specifies the name of a temporary sequential data set in which to log output from the DLA TSO FTP client. The file that you specify is used to buffer the output from the TSO FTP client. This sequential file exists only while the DLA function is running.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_space_pri

Purpose

The DLA.ftp_log_space_pri statement is used by the Discovery Library Adapter (DLA) to define the space allocation (primary extents) for the file specified by the DLA.ftp_log_filename statement.

Syntax

The DLA.ftp_log_space_pri statement has the following syntax:

DLA.ftp_log_space_pri

```
►►[DLA.ftp_log_space_pri = 150]►►  
►►DLA.ftp_log_space_pri = ftp_log_primary►►
```

where:

ftp_log_primary

The number of primary extents (in cylinders) allocated to the file defined by the DLA.ftp_log_filename statement. The default value is 150 cylinders.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_space_sec

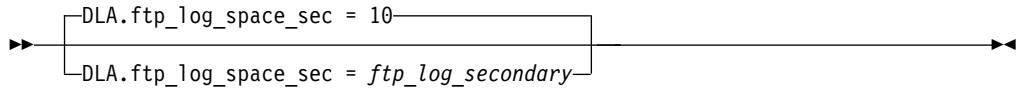
Purpose

The DLA.ftp_log_space_sec statement is used by the Discovery Library Adapter (DLA) to define the space allocation (secondary extents) for the file specified by the DLA.ftp_log_filename statement.

Syntax

The DLA.ftp_log_space_sec statement has the following syntax:

DLA.ftp_log_space_sec



where:

ftp_log_secondary

The number of secondary extents (in cylinders) allocated to the file defined by the DLA.ftp_log_filename statement. The default value is 10 cylinders.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_unit

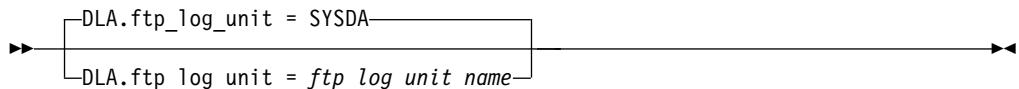
Purpose

The DLA.ftp_log_unit statement is used by the Discovery Library Adapter (DLA) to define the UNIT parameter to be used during the allocation of DLA.ftp_log_filename. This parameter can contain the group-name , device-type, or device-number.

Syntax

The DLA.ftp_log_unit statement has the following syntax:

DLA.ftp_log_unit



where:

ftp_log_unit_name

Specifies the UNIT parameter to be used for the file defined by the DLA.ftp_log_filename statement. This parameter can contain one of the following values:

- group name (for example, SYSDA)
- device-type (for example, 3390)
- device-number (for example /2301)

The default value is SYSDA.

Usage notes

- This statement is optional.

- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_log_volume

Purpose

The DLA.ftp_log_volume statement is used by the Discovery Library Adapter (DLA) to define the volume on which the file specified by the DLA.ftp_log_filename statement will be created.

Syntax

The DLA.ftp_log_volume statement has the following syntax:

DLA.ftp_log_volume

►►—DLA.ftp_log_volume = *volume*—————►►

where:

volume

Specifies the volume on which the file specified by DLA.ftp_log_filename is created.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_max_xmit_tm

Purpose

The DLA.ftp_max_xmit_tm statement is used by the Discovery Library Adapter (DLA) to define the maximum amount of time for the transfer of data to the Configuration Management Database (CMDB) FTP server. If you have a slow link between the operating system and the CMDB FTP server or if the DLA book is very large, you can increase this value.

Syntax

The DLA.ftp_max_xmit_tm statement has the following syntax:

DLA.ftp_max_xmit_tm

►►—

DLA.ftp_max_xmit_tm = 600
DLA.ftp_max_xmit_tm = <i>max_seconds</i>

—————►►

where:

max_seconds

Specifies the maximum amount of time, in seconds, for transferring the XML discovery book to the CMDB FTP server. The default value is 600 seconds.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_remote_dir

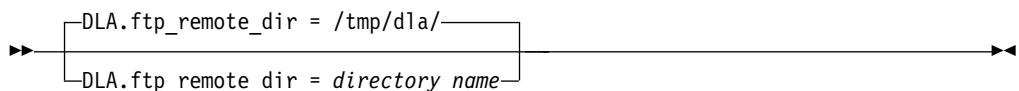
Purpose

The DLA.ftp_remote_dir statement is used by the Discovery Library Adapter (DLA) to define the destination directory on the Configuration Management Database (CMDB) FTP server where the DLA book is placed.

Syntax

The DLA.ftp_remote_dir statement has the following syntax:

DLA.ftp_remote_dir



where:

directory_name

Specifies the directory on the CMDB FTP server to which the XML discovery book is transferred. The default value is `/tmp/dla/`. The ending slash (/) is required.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_timeout

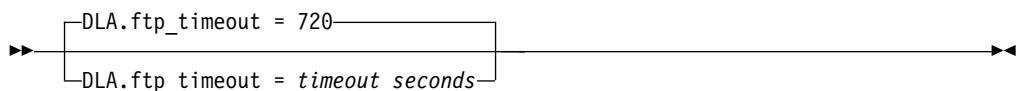
Purpose

The DLA.ftp_timeout statement is used by the Discovery Library Adapter (DLA) to define the maximum amount of time to wait before an FTP connection is established to the Configuration Management Database (CMDB) FTP server.

Syntax

The DLA.ftp_timeout statement has the following syntax:

DLA.ftp_timeout



where:

timeout_seconds

Specifies the FTP connection timeout value, which is specified in seconds. The default value is 720 seconds.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.ftp_uid

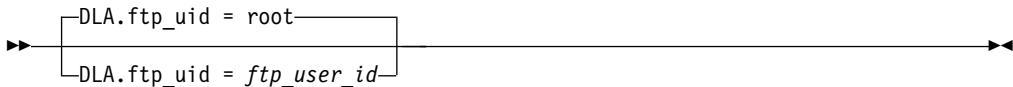
Purpose

The DLA.ftp_uid statement is used by the Discovery Library Adapter (DLA) to define the user ID with which the TSO FTP client connects to the Configuration Management Database (CMDB) FTP server.

Syntax

The DLA.ftp_uid statement has the following syntax:

DLA.ftp_uid



where:

ftp_user_id

Specifies the user ID on the CMDB FTP server. Define the password for this user ID using the PWD.DLA.P statement in the CNMSTPWD member. The default value is root.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.is_second_level

Purpose

The DLA.is_second_level statement is used by the Discovery Library Adapter (DLA) to indicate that the NetView program is running on a second-level system (the z/OS system is running in a z/VM® virtual machine).

Syntax

The DLA.is_second_level statement has the following syntax:

DLA.is_second_level

```
►►—DLA.is_second_level = [0] [1]—►►
```

where:

- 0 Specifies that the NetView program is not running on a second-level system. This is the initial setting in the CNMSTYLE member.
- 1 Specifies that the NetView program is running on a second-level system.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.statefile

Purpose

The DLA.statefile statement is used by the Discovery Library Adapter (DLA) to define the name of the file that contains certain state-based information used by the DLA adapter.

Syntax

The DLA.statefile statement has the following syntax:

DLA.statefile

```
►►—DLA.statefile = state_file_name—►►
```

where:

state_file_name

Defines a sequential data set on the local system that contains state information for the DLA adapter.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.tsouser

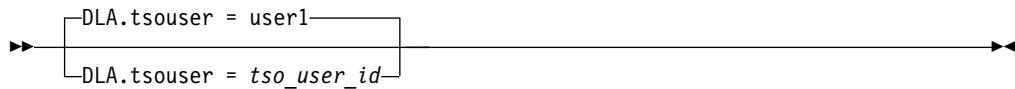
Purpose

The DLA.tsouser statement is used by the Discovery Library Adapter (DLA) to define the user ID with which to start the NetView TSO server. For more information about this user ID, see the online help for the NCCF START command (section on TSOUSER).

Syntax

The DLA.tsouser statement has the following syntax:

DLA.tsouser



where:

tso_user_id

Specifies the user ID that the TSO command server (TSOSERV) function runs to perform the FTP file transfer. The default value is user1.

Usage notes

- This statement is required if you are using the DLA function.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_blocksize

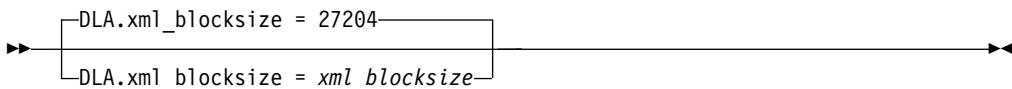
Purpose

The DLA.xml_blocksize statement is used by the Discovery Library Adapter (DLA) to specify the block size and average record length for a data control block (DCB) used during the allocation of the XML discovery book.

Syntax

The DLA.xml_blocksize statement has the following syntax:

DLA.xml_blocksize



where:

xml_blocksize

The blocksize can be a value from 0 - 65536. The default value is 27204.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_filename

Purpose

The DLA.xml_filename statement defines where to store the XML discovery book on the local z/OS system.

Syntax

The DLA.xml_filename statement has the following syntax:

DLA.xml_filename

►—DLA.xml_filename = *xml_file_name*————→

where:

xml_file_name

Specifies the name of a sequential data set in which to store the XML discovery book. The data set is retained until the next time that the DLA runs. This data set cannot be a partitioned data set.

Usage notes

- This statement is required if you are using the DLA function.
- If you change the *xml_file_name* parameter between DLA runs, any data sets that you specified earlier and were written to by the DLA will not be deleted.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_space_pri

Purpose

The DLA.xml_space_pri statement defines the space allocation (primary extents) for the file specified by the DLA.xml_filename statement.

Syntax

The DLA.xml_space_pri statement has the following syntax:

DLA.xml_space_pri

►—DLA.xml_space_pri = 150
 |————DLA.xml_space_pri = *xml_primary*————→

where:

xml_primary

This defines the primary extents, specified in cylinders. The default value is 150 cylinders.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_space_sec

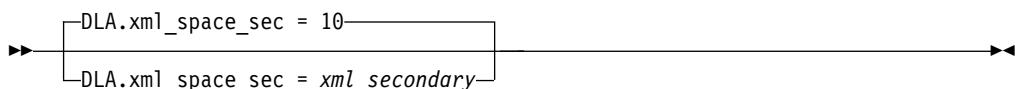
Purpose

The DLA.xml_space_sec statement defines the space allocation (secondary extents) for the file specified by the DLA.xml_filename statement.

Syntax

The DLA.xml_space_sec statement has the following syntax:

DLA.xml_space_sec



where:

xml_secondary

This defines the secondary extents, specified in cylinders. The default value is 10 cylinders.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_unit

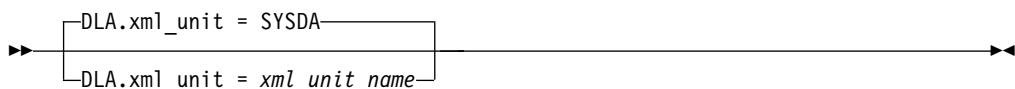
Purpose

The DLA.xml_unit statement is used by the Discovery Library Adapter (DLA) to define the UNIT parameter for the log specified by the DLA.xml_filename statement. This parameter can contain the group-name , device-type, or device-number.

Syntax

The DLA.xml_unit statement has the following syntax:

DLA.xml_unit



where:

xml_unit_name

Specifies the UNIT parameter to be used for the file defined by the DLA.xml_filename statement. This parameter can contain one of the following values:

- group name (for example, SYSDA)

- device-type (for example, 3390)
- device-number (for example /2301)

The default value is SYSDA.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DLA.xml_volume

Purpose

The DLA.xml_log_volume statement is used by the Discovery Library Adapter (DLA) to define the volume on which the file specified by the DLA.xml_filename statement will be created.

Syntax

The DLA.xml_volume statement has the following syntax:

DLA.xml_volume

►►—DLA.xml_volume = *volume*—————►►

where:

volume

Specifies the volume on which the file specified by DLA.xml_filename is created.

Usage notes

- This statement is optional.
- Additional information about the DLA function can be found in the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

DOMAIN

Purpose

The DOMAIN statement specifies the NetView domain name as a global variable.

Syntax

The DOMAIN statement has the following syntax:

DOMAIN

►►—

DOMAIN = C&NV2I.01
DOMAIN = <i>domainid</i>

—————►►

where:

domainid

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

Usage notes

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

DVIPA.Event.Delay

Purpose

The DVIPA.Event.Delay statement specifies the number of seconds that each NetView program waits after a z/OS Communications Server event (DVIPA SNMP trap, DVIPA TCP/IP profile update, or sysplex monitoring message) is received before performing the following actions:

- Rediscovering DVIPA information for this z/OS system if the NetView program is not running in an environment with a master NetView program.
- Notifying the master NetView program that rediscovery of DVIPA information is required for this z/OS system.

Syntax

The DVIPA.Event.Delay statement has the following syntax:

DVIPA.Event.Delay



where:

seconds

Specifies the time in the range of 0 - 600 seconds. The default value is 60 seconds.

Usage notes

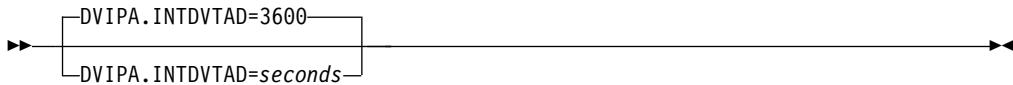
- If you set the value of the DVIPA.Event.Delay statement to 0, no delay occurs. Depending on which DVIPA events that you are monitoring, you can receive multiple events in a short time span, which would lead to rediscovery for each individual event.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.INTDVTAD

The DVIPA.INTDVTAD statement defines the time interval for data collectors to collect distributed DVIPA targets, DVIPA sysplex distributors, and distributed DVIPA server health data.

The DVIPA.INTDVTAD statement has the following syntax:

DVIPA.INTDVTAD



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 3600 seconds.

Usage Notes:

- The specified interval on the DVIPA.INTDVTAD statement also controls how often data is collected for distributed DVIPA statistics. For more information, see *IBM Tivoli NetView for z/OS IP Management*.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.INTDVCONN

The DVIPA.INTDVCONN statement defines the time interval for data collectors to collect DVIPA connections data.

The DVIPA.INTDVCONN statement has the following syntax:

DVIPA.INTDVCONN



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 3600 seconds.

Usage Note: To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.INTDVDEF

The DVIPA.INTDVDEF statement defines the time interval for data collectors to collect DVIPA definition and status data.

The DVIPA.INTDVDEF statement has the following syntax:

DVIPA.INTDVDEF



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 3600 seconds.

Usage Notes:

- Application-Instance DVIPA data and Stack-Defined DVIPA data are subsets of the DVIPA Definition and Status data and are therefore also collected on the DVIPA.INTDVDEF interval.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.INTDVROUT

The DVIPA.INTDVROUT statement defines the time interval for data collectors to collect VIPA routes and distributed DVIPA connection routing data.

The DVIPA.INTDVROUT statement has the following syntax:

DVIPA.INTDVROUT



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 3600 seconds.

Usage Note:

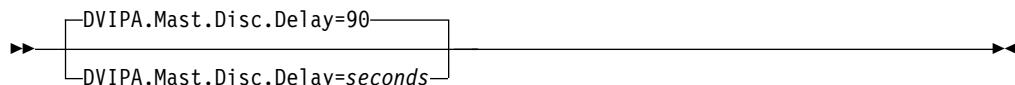
To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.Mast.Disc.Delay

The DVIPA.Mast.Disc.Delay statement specifies the number of seconds that the master NetView program waits to receive additional notifications from systems in the sysplex where a DVIPA event occurred, and for which rediscovery is required. This delay interval starts when the first notification is received indicating that an event triggered the need for rediscovery.

The DVIPA.Mast.Disc.Delay statement has the following syntax:

DVIPA.Mast.Disc.Delay



where:

seconds

Specifies the time in the range of 0 - 600 seconds. The default value is 90 seconds.

Usage Notes:

- If you set the value of the DVIPA.Mast.Disc.Delay statement to 0, no delay occurs. Depending on which DVIPA events that you are monitoring, and how widespread the change is in the DVIPA data in the sysplex, you can receive multiple events in a short time span, which can lead to rediscovery for each individual event.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.Mast.EMARf.Delay

The DVIPA.Mast.EMARf.Delay statement specifies the time period that the master NetView program waits before writing data to the NetView for z/OS Enterprise Management Agent data space. By using this statement, multiple updates to a data space can occur with one operation. The delay starts when the first notification is received indicating that new DVIPA data was received by the master NetView program. When the delay expires, the data is available for viewing in the NetView for z/OS Enterprise Management Agent DVIPA workspaces.

The DVIPA.Mast.EMARf.Delay statement has the following syntax:

DVIPA.Mast.EMARf.Delay



where:

seconds

Specifies the time in the range 0-600 seconds. The default value is 180 seconds.

Usage Note: To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.DVIPA

The DVIPA.STATS.DVIPA statement specifies the distributed DVIPA or set of distributed DVIPAs for which records are to be written.

The DVIPA.STATS.DVIPA statement has the following syntax:

DVIPA.STATS.DVIPA



where:

value

Specifies the distributed DVIPA in the following formats:

- An asterisk (*) specifies that all distributed DVIPAs will be written.
- An IPv4 address in dotted-decimal format: *ddd.ddd.ddd.ddd*.

Each *ddd* can be any of the following formats:

- A decimal number 0 - 255
- A hyphen-separated range (for example, 240 - 255)
- An asterisk (*) represents the range 0 - 255

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

- An IPv6 address in colon-hexadecimal format:

hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh:hhhh, where each *hhhh* is a 0 - 4 digit hexadecimal number, or a hexadecimal range separated by a hyphen, for example the range FF00 - FFFF.

Consecutive groups of zeros can be replaced with a double colon (:). A double colon can be used to signify leading, trailing or embedded groups of zeros, and can be specified only once in an address.

A single asterisk (*) can be used in place of *hhhh* to denote a hexadecimal range of 0 - FFFF. If the last *hhhh* is an asterisk and less than 8 *hhhh* values are specified, a range of 0 - FFFF is used for each remaining *hhhh*. If both an asterisk and a double colon are used in an address, the asterisk only represents a single *hhhh* group regardless of its position.

If an IPv6 address is specified in a mixed format (containing both colons and dots), then the NetView program inspects the high-order 12 bytes of the 16-byte address. If the first 12 bytes of the 16-byte address is an IPv4 migration value of 0:0:0:0:FFFF or 0:0:0:0:0, then the NetView program strips the high-order 12 bytes and processes the remaining 4 bytes as an IPv4 address of the dotted decimal format *ddd.ddd.ddd.ddd*. The following examples are processed as IPv4 addresses:

```

::*
::FFFF:*
9.42.44.52
::FFFF:9.42.44.52
00:000:0000:0:000:FFFF:9.42.44.52
::0:0:FFFF:9.42.44.52

```

Usage Notes:

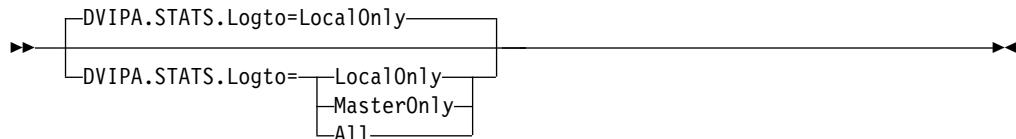
- The value for DVIPA is assigned in the following way:
 1. The DVIPA value specified with the DVIPALOG command.
 2. The DVIPA.STATS.DVIPA variable in the CNMSTYLE member.
 3. The default value of asterisk (*).
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.Logto

The DVIPA.STATS.Logto statement specifies the logging option for distributed DVIPA statistical records for master-capable and basic NetView programs.

The DVIPA.STATS.Logto statement has the following syntax:

DVIPA.STATS.Logto



where:

LocalOnly

Specifies to log the records locally and not forward records to the master NetView program. This is the default value.

MasterOnly

Specifies to forward records to the master NetView program and not log the records locally.

All

Specifies to log the records locally and forward the records to the master NetView program.

Usage Notes:

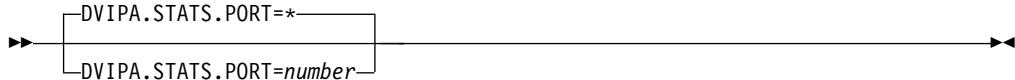
- You can turn the logging function off by setting the Init.DVIPASTATS statement in the CNMSTYLE member to No. Initially, logging is off.
- At the master NetView program, the DVIPA.STATS.Logto statement is ignored. Local and forwarded records are logged, unless the logging function is off.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.PORT

The DVIPA.STATS.PORT statement specifies the distributed DVIPA port for which records are to be written.

The DVIPA.STATS.PORT statement has the following syntax:

DVIPA.STATS.PORT



where:

number

Defines the port number. The default is asterisk (*), all ports.

Usage Notes:

- The value for PORT is assigned in the following way:
 1. The PORT value specified with the DVIPALOG command.
 2. The DVIPA.STATS.PORT variable in the CNMSTYLE member.
 3. The default value of asterisk (*).
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.Pri.MAXR

The DVIPA.STATS.Pri.MAXR statement specifies how many records to write to the primary data set. When this amount is reached, logging switches to the secondary data set if it is available.

The DVIPA.STATS.Pri.MAXR statement has the following syntax:

DVIPA.STATS.Pri.MAXR



where:

number

Specifies a number from 100 - 500000. The default value is 500.

Usage Notes:

- If the *number* specified is not valid, the default value of 500 is used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.Sec.MAXR

Purpose

The DVIPA.STATS.Sec.MAXR statement specifies how many records to write to the secondary data set. When this amount is reached, logging switches to the primary data set if it is available.

Syntax

The DVIPA.STATS.Sec.MAXR statement has the following syntax:

DVIPA.STATS.Sec.MAXR



where:

number

Specifies a number from 100 - 500000. The default value is 500.

Usage notes

- If the *number* specified is not valid, the default value of 500 is used.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

DVIPA.STATS.TCPNAME

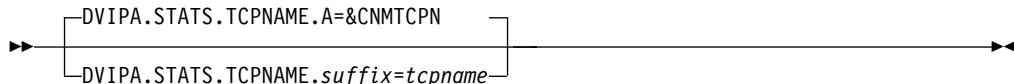
Purpose

The DVIPA.STATS.TCPNAME statement specifies the local TCP/IP job name that is associated with distributed DVIPA data for which records are to be written. You can specify multiple TCPNAME statements.

Syntax

The DVIPA.STATS.TCPNAME statement has the following syntax:

DVIPA.STATS.TCPNAME



where:

suffix

Defines additional DVIPA.STATS.TCPNAME statements. The suffix consists of 1 - 8 characters and must be unique (A and B in the example).

(DVIPA)DVIPA.STATS.TCPNAME.A = &CNMTCPN
*DVIPA.STATS.TCPNAME.B = tcpname2

tcpname

Specifies the local TCP/IP job name associated with the distributed DVIPA data.

Usage notes

- The value for TCPNAME is assigned in the following way:
 1. The TCPNAME value or values specified with the DVIPALOG command.
 2. The DVIPA.STATS.TCPNAME.*suffix* variables in the CNMSTYLE member.
 3. The &CNMTCPPN NetView system symbolic, the value for which comes from the TCPNAME parameter in the CNMSTYLE member. This value is always present because CNMSTYLE processing assigns the value TCPIP if no TCPNAME statement is coded in the CNMSTYLE member.
 4. The default value of TCPIP.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE DVIPA command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

EMAAUTO

Purpose

The EMAAUTO statement specifies whether the Tivoli NetView for z/OS Enterprise Management Agent or the Tivoli Enterprise Monitoring Server is to be started during NetView initialization. The TEMA tower must be enabled.

Syntax

The EMAAUTO statement has the following syntax:

EMAAUTO

►►—EMAAUTO. [TEMA] .procStr = [procedure_cmdstr] —►►

where:

procedure_cmdstr

Specifies the startup procedure and startup parameters.

NONE

Specifies that the *procedure_name* should not be started.

TEMA.procStr

Specifies the NetView for z/OS Enterprise Management Agent startup procedure. If you do not want the NetView program to start this procedure, specify *NONE* for *procedure_name*.

TEMS.procStr

Specifies the Tivoli Enterprise Monitoring Server startup procedure name. If you do not want the NetView program to start this procedure, specify *NONE* for *procedure_name*.

Usage notes

- The TEMA tower must be enabled.

- The INIT.EMAAUTO statement must be set to Yes for the EMAAUTO.TEMA.procStr and EMAAUTO.TEMS.procStr statements to be processed.
- The NetView program only processes one startup procedure statement. If you specify a value other than *NONE* for EMAAUTO.TEMS.procStr statement, the Tivoli Enterprise Monitoring Server is started. When successfully started, the NetView message automation starts the NetView for z/OS Enterprise Management Agent by starting the procedure specified with the EMAAUTO.TEMA.procStr statement. See the CNMSEMAA automation member for more information. You can specify parameters for the Tivoli Enterprise Monitoring Server startup procedure by adding them to the EMAAUTO.TEMS.procStr statement.
- If you only want the NetView for z/OS Enterprise Management Agent to be started, specify *NONE* on the EMAAUTO.TEMS.procStr statement and specify the name of the startup procedure on the EMAAUTO.TEMA.procStr statement. You can specify parameters for the NetView for z/OS Enterprise Management Agent startup procedure by adding them to the EMAAUTO.TEMA.procStr statement.

endcmd.AutoTask

Purpose

The endcmd.AutoTask statement defines the autotask that runs commands that are specified on the PIPE KEEP stage using the GLOBAL and ENDCMD parameters.

Syntax

The endcmd.AutoTask statement has the following syntax:

endcmd.AutoTask

►—endcmd.AutoTask—=autotask_name—————►

where:

autotask_name

Specifies a valid operator ID or a function name defined on a function.autotask CNMSTYLE statement. The initial value is ?PRIMARY.

Usage note

If the NetView program shuts down because of a CLOSE IMMED or CLOSE STOP command, the specified task continues to process new commands from the global keeps for a time not exceeding that specified by the endcmd.close.leeway statement.

endcmd.close.leeway

Purpose

The endcmd.close.leeway statement specifies how long commands specified using the ENDCMD parameter on the PIPE KEEP stage can run after a CLOSE IMMED, CLOSE STOP, or an MVS STOP (P) command is entered for the NetView program. The total for all commands cannot exceed the specified leeway value. This

statement applies only to the global keeps.

Syntax

The endcmd.close.leeway statement has the following syntax:

endcmd.close.leeway

►—endcmd.close.leeway=*time*—►

where:

time

Specifies the number of seconds. The initial value is 2 seconds.

Usage notes

- The leeway value does not apply to CLOSE ABEND command processing or when an MVS CANCEL command is used to end the NetView program.
- Choosing a large leeway number can be disruptive. Because of this, specify the smallest number of seconds such that anticipated commands can complete processing.

ENT.CONNCHECK.INT

Purpose

The ENT.CONNCHECK.INT statement specifies the interval for how often a sysplex master or enterprise master NetView program verifies RMTCMD connectivity to other systems in the sysplex or enterprise.

- For sysplex masters, connectivity is checked to other systems in the sysplex.
- For enterprise masters, connectivity is checked to systems that are specified on ENT.SYSTEMS statements and that were contacted. If a contacted system is in a sysplex, connectivity to other systems in the sysplex is also checked.

Syntax

The ENT.CONNCHECK.INT statement has the following syntax:

ENT.CONNCHECK.INT

►—ENT.CONNCHECK.INT = 1—
 |
 ENT.CONNCHECK.INT = *minutes*—►

where:

minutes

Specifies the number of minutes (1 - 60). The default value is 1 minute.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ENT command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

ENT.GROUP.*groupname*

Purpose

The ENT.GROUP.*groupname* statement defines a group of local or remote NetView instances. You can use a group to define a logical cluster of NetView instances; you can then use the group with the BROWSE command to see data from all NetView instances in the cluster. A group can include specific NetView domains, sysplexes, and other groups.

Syntax

The ENT.GROUP.*groupname* statement has the following syntax:

ENT.GROUP.*groupname*



where:

groupname

Specifies the 1-character through 20-character name of the group. The group name can contain alphanumeric characters and the special characters @, #, and \$.

grouplist

Specifies one or more destinations for the BROWSE command, separated by spaces. Each destination can be any of the following:

- A NetView domain name
- An alias defined using the RMTSYN and RMTALIAS statements
- The local sysplex name
- A sysplex name discovered by the NetView program as a result of ENT.SYSTEMS.*name* statement processing
- The name of another group defined using the ENT.GROUP.*groupname* statement

Usage notes

- Groups cannot include their own group names in the definition.
 - You can browse remote sysplexes from the enterprise master NetView program when the remote sysplexes belong to the same enterprise. You can browse a remote sysplex on a sysplex that has the same enterprise master as the local NetView program. This local NetView program does not have to be an enterprise master.
 - You can determine the members of a local sysplex by using **LIST STATUS=XCFGRPS** command. To determine the members of a remote sysplex, use the **LIST STATUS=XCFGRPS** command on the enterprise master. The list of members can be affected by configuration on the remote systems and the state of enterprise master processing.
- You can browse members using the **BROWSE** command.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ENT command.

Examples

- **Example 1:** Group 4 is comprised of group 1, group 2, and group 3 definitions.

```
ENT.GROUP.group1 = domid rmtalias rmstsyn
ENT.GROUP.group2 = group1 rmstsyn2 rmstsyn3
ENT.GROUP.group3 = entsys1 entsys2
ENT.GROUP.group4 = group2 group3
```

ENT.INT.name

Purpose

The ENT.INT.*name* statement specifies the number of seconds to wait before sending discovery commands from the enterprise master NetView program to other systems defined by ENT.SYSTEMS.*name* statements.

The *name* specification must match the system or sysplex *name* specified on the ENT.SYSTEMS.*name* statement. When a discovery command is sent to the specified system, or one of the systems in that sysplex, the NetView program will wait for the interval specified by the ENT.INT.*name* statement before sending any other discovery commands to managed systems outside its sysplex.

Syntax

The ENT.INT.*name* statement has the following syntax:

ENT.INT.name



```
►—ENT.INT.name [ .sysname] =seconds —►
```

where:

name

Entry point into a sysplex or standalone system. This value must be the same as the name specified on the ENT.SYSTEMS.*name* statement.

sysname

Specifies a particular NetView domain within a sysplex.

seconds

A value between 0 and 3600 that specifies the number of seconds to wait. The default value is 60 seconds.

Usage notes

- You can use the ENT.INT.*name* statement to specify the time to wait before doing data discovery when you are switching to another enterprise master NetView program.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ENT command.

Examples

- **Example 1:** CALPLEX sysplex

```
ENT.INT.CALPLEX = 30
```

When the discovery command for the CALPLEX sysplex is sent out, the enterprise master NetView program waits 30 seconds before sending any other discovery commands to managed systems within the enterprise. This interval applies to each system in the sysplex.

- **Example 2:** Specific domain (CNM01) in the CALPLEX sysplex

```
ENT.INT.CALPLEX.CNM01 = 45
```

When the discovery command for the CNM01 domain in the CALPLEX sysplex is sent out, the enterprise master NetView program waits 45 seconds before sending other discovery commands to the systems within the sysplex.

- **Example 3:** ONESYS system

```
ENT.INT.ONESYS = 120
```

The enterprise master NetView program waits 120 seconds before sending other discovery commands to other systems or sysplexes that are defined on the ENT.SYSTEMS.*name* statements.

ENT.SYSTEMS.*name*

Purpose

The ENT.SYSTEMS.*name* statement specifies entry points into XCF groups in sysplex systems or stand-alone systems that can be managed.

An entry point into a sysplex is a NetView domain that returns information about other NetView domains in its XCF group in the sysplex to the enterprise master. This enables the enterprise master to discover all domains in a sysplex without requiring configuration for contacting each domain. Only configuration for contacting the entry point or points (a RMTALIAS definition) is required.

To provide backup entry points, you can specify more than one entry point on the ENT.SYSTEMS.*name* statement. If you do not specify a backup entry point, you can lose connectivity to remote systems.

Syntax

The ENT.SYSTEMS.*name* statement has the following syntax:

ENT.SYSTEMS.*name*



where:

name

Specifies a label for the entry point into a sysplex or stand-alone system.

alias

Specifies one or more defined RMTCMD aliases that you can use to contact a

sysplex member or stand-alone system. In the case of a sysplex, additional aliases (to additional NetView domains) can serve as backup entry points into the sysplex. The NetView program processes the aliases specified on the statement from left to right until one of the entry points is contacted.

For additional information on how to specify the RMTCMD alias, see the RMTALIAS and RMTSYN statements.

Usage notes

- The enterprise master NetView program goes through the aliases in sequence until it successfully contacts a sysplex member or stand-alone system, or the list is ended.
- If DVIPA connectivity exists to the sysplex and the sysplex is using the XCF.MASTDVIPA statement to define a DVIPA address for the master NetView program, you can specify an alias for the DVIPA address of the master NetView program. This alias can be used to reach the master NetView program regardless of its location in the sysplex. By using the ENT.SYSTEMS.*name* statement, you can reduce the number of aliases that are required to contact a system in the sysplex.
- Make sure that the RMTALIAS and RMTSYN statements define the aliases that you specify on the ENT.SYSTEMS.*name* statement.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE ENT command.

Examples

- An example ENT.SYSTEMS.*name* statement follows for the CALPLEX sysplex that specifies TAMPA and SANDIEGO as RMTCMD aliases:

ENT.SYSTEMS.CALPLEX = TAMPA SANDIEGO

These aliases must also be specified on the RMTCMD statement:

RMTALIAS.SANDIEGO = LU62.CNM01
RMTALIAS.TAMPA = USIBMNT.CNM02

- An example ENT.SYSTEMS.*name* statement follows for the ONESYS system.

ENT.SYSTEMS.ONESYS = SPV6

function.autotask

Purpose

The function.autotask statement defines the autotask for the specified function. Refer to the CNMSTYLE member for specific information on the function.autotask statements.

Syntax

The function.autotask statement has the following syntax:

function.autotask

►—function.autotask.—*function_name*=*autotask_name*————►

where:

autotask_name

Specifies the autotask name. The autotask name must be 1 - 8 alphanumeric characters in length; can contain no special characters except #, @, and \$; and must start with either an alphabetic character or #, @, or \$.

function_name

Specifies the function name. The function name can be 1 - 15 characters and can contain letters, numbers, and periods. The first character must be a letter.

You can prefix *function_name* with a question mark (?) and then use this in other areas of the NetView program. This allows the function name to be used in many places where an operator ID is required. For example, you can use ?*function_name* on the AUTOTASK statement , the QOS command, and the AUTOTASK command.

Usage notes

- For a list of autotasks and their functions, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- The function.autotask.primary statement provides a set of related processes, possibly including timed commands, message automation, data services task (DST) services, and control (operator) commands. This statement is also the autotask that is used when an error is encountered with the task name on another function.autotask statement. Because of this, the function.autotask.primary statement is required and cannot contain a value of *NONE*.
- The following statement is used by MultiSystem Manager:
(MSM) function.autotask.MSMdefault = AUTOMSMD

The MSMdefault autotask is used if the AUTOTASK keyword is not specified on the GETTOPO command and the related service point object is not in RODM.

- For any autotasks that you define to the NetView program, also consider defining them to the SAF product. The automation table and other NetView functions use these definitions.

GHB.TCPANAME

Purpose

Use the GHB.TCPANAME statement to specify the TCP name for the DUIDGHB task.

Syntax

The GHB.TCPANAME statement has the following syntax:

GHB.TCPANAME

►►—GHB.TCPANAME— = —*name*—►►

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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE GHB command.

HARDCOPY

Purpose

The HARDCOPY statement designates the devices that you can use for hardcopy printers. Code one HARDCOPY statement and continue it on multiple lines as needed to define your printers. For the changes to take effect, stop and restart the NetView program.

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 might accumulate, and might not be printed for some time after they are received.

Syntax

The HARDCOPY statement has the following syntax:

HARDCOPY



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. Separate values using blanks.

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 M3287SCS 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 default supplied by the NetView program (3000) applies to hardcopy printers. Hardcopy printers can become backlogged if they are slow or out of paper.

Related statements

NORMQMAX

HLLENV

Purpose

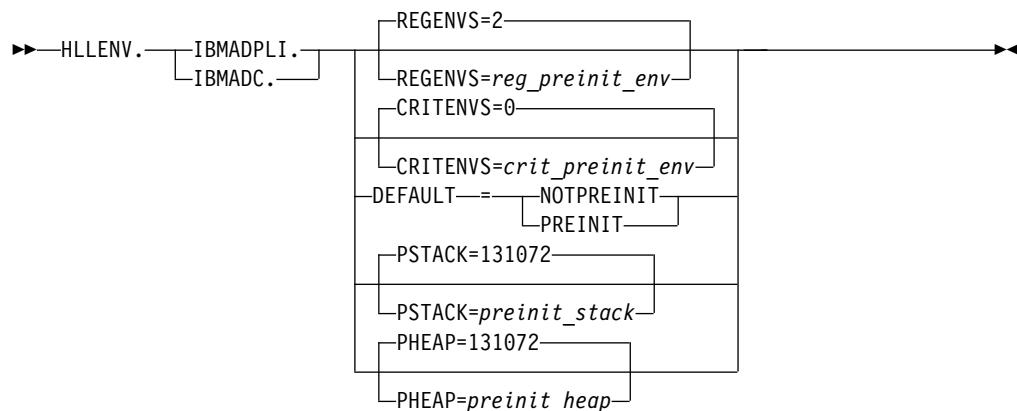
The HLLENV statements define the following preinitialized environments:

- PL/I
- C

Syntax

The HLLENV statement has the following syntax:

HLLENV



where:

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 HLOOPTS. 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 is the default.

NOTPREINIT

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

PREINIT

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

IBMADPLI

Specifies the PL/I environment.

IBMADC

Specifies the C environment.

PHEAP=preinit_heap

Specifies the HEAP runtime option value used when building the preinitialized environment. The initial value for PHEAP is 131072 bytes.

PSTACK=preinit_stack

Specifies the STACK runtime option value used when building the preinitialized environment. The initial value for PSTACK is 131072 bytes.

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 pre-initialized environment. These programs can run in a preinitialized environment on any subtask where the program ordinarily runs. The default value is 2.

Usage notes

- For more information, refer to the online help for the HLLENV command.
- Refer to *IBM Tivoli NetView for z/OS Programming: PL/I and C* for more information about HLLOPTS.

idleParms

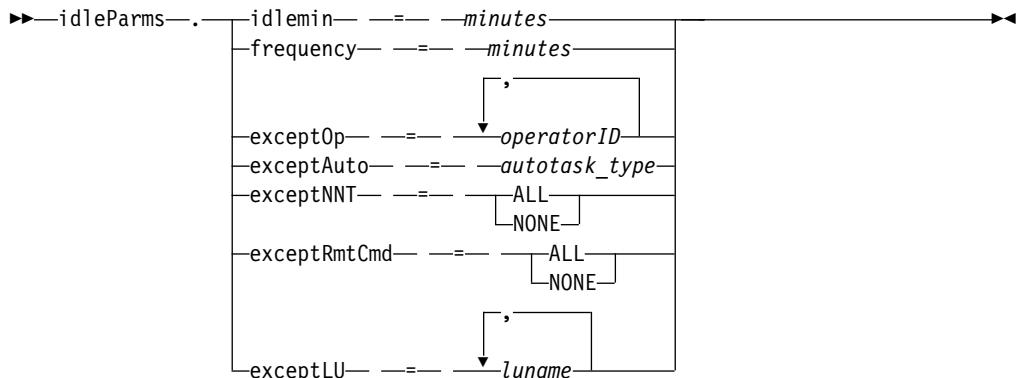
Purpose

The idleParms statement specifies operator idle times that are used to determine which operators are to be stopped.

Syntax

The idleParms statement has the following syntax:

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 called. You can specify a value in the range of 0.0167 (equivalent to one second) - 500000. Do not specify either a comma or a space in the value that you specify.

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
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
ALL Specifies that all NNT tasks are exempt.

NONE
 Specifies that all NNT tasks are eligible. This is the initial setting.

exceptRmtCmd
 Specifies which RMTCMD users are exempt from IDLEOFF.

ALL
ALL Specifies that all users of RMTCMD are exempt, but does not exempt distributed autotasks.

Note: 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.

NONE
 Specifies that all RMTCMD users are eligible. This is the initial setting.

exceptLU = *luname*
 Specifies a list of LU names that are not to be logged off.

Usage note

You can also use the IDLEOFF command to examine operator idle times. For more information, see the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* or the NetView online help.

IDS.Attack_Cmd

Purpose

The IDS.Attack_Cmd statement specifies the command to be issued when an IDS attack detection event occurs. Code as many of these statements as needed for your environment. The IDS.Attack_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Attack_Cmd statement has the following syntax:

►—IDS.—Attack—Cmd.—*suffix*—=—*command*—►

where:

suffix

A numerical suffix. Increment each suffix by one for each command used. In the example that follows, two commands, a UNIX command and a NetView command, are defined.

```
IDS.Attack_CmdType.1 = UNIX  
IDS.Attack_Cmd.1 = /bin/trmdstat -A -D IDS.SYSLOG.FILENAME  
IDS.Attack_CmdType.2 = NETV  
IDS.Attack_Cmd.2 = WHO
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Attack_Cmd.4 statement is ignored:

```
IDS.Attack_Cmd.1  
IDS.Attack_Cmd.2  
IDS.Attack_Cmd.4
```

command

Specifies the command to be issued.

Usage notes

The IDS.Attack_CmdType statement specifies the command environment.

Related statements

IDS.Attack_CmdType

IDS.Attack_CmdType

Purpose

The IDS.Attack_CmdType statement defines the command environment to be used when an IDS attack detection event occurs. Code as many of these statements as needed for your environment. The IDS.Attack_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Attack_CmdType statement has the following syntax:

►—IDS.—Attack—CmdType.—*suffix*=*UNIX*—►

►—IDS.—Attack—CmdType.—*suffix*=—►

 |

 | MVS

 |

 | NETV

 |

 | UNIX

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, two command environments, UNIX and NETV, are defined.

```
IDS.Attack_CmdType.1 = UNIX  
IDS.Attack_Cmd.1 = /bin/trmdstat -A -D IDS.SYSLOG.FILENAME  
IDS.Attack_CmdType.2 = NETV  
IDS.Attack_Cmd.2 = WHO
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Attack_CmdType.3 statement is ignored:

```
IDS.Attack_CmdType.1 = UNIX  
IDS.Attack_CmdType.3 = NETV
```

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage notes

- The IDS.Attack_Cmd statements define the commands to be issued.

Related statements

IDS.Attack_Cmd

IDS.Auto_Intvl

Purpose

The IDS.Auto_Intvl statement defines the time interval that you use when defining IDS event threshold values. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Auto_Intvl statement has the following syntax:

IDS.Auto_Intvl



where:

hh:mm:ss

Indicates the interval in hours (24-hour clock), minutes, and seconds for an IDS event threshold. The initial default value is 5 minutes (00:05:00).

Related statements

IDS.Auto_Thresh

IDS.Auto_Thresh

Purpose

The IDS.Auto_Thresh statement defines the automation table threshold value for the number of IDS events. When the threshold value is exceeded, no event reports are generated. The IDS.Auto_Thresh statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Auto_Thresh statement has the following syntax:

IDS.Auto_Thresh

►—IDS.Auto_Thresh—=—nnn—►

where:

nnn

Indicates the threshold for the number of IDS events. The number must be in the range of 1 - 999. The initial setting is 100.

Usage notes

- The IDS.Auto_Thresh and IDS.Auto_Intvl statements set the IDS event threshold used by the automation table. In the following example, if 101 or more IDS events are detected within 5 minutes, no IDS event reports are generated:

```
IDS.Auto_Thresh = 100  
IDS.Auto_Intvl = 00:05:00
```

Related statements

IDS.Auto_Intvl

IDS.ClearStat_Day

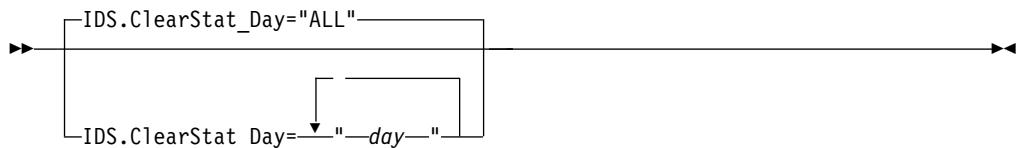
Purpose

The IDS.ClearStat_Day statement defines when to clear the probe summary statistics. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.ClearStat_Day statement has the following syntax:

IDS.ClearStat_Day



where:

ALL

Indicates that the probe summary statistics are to be cleared every day. This is the default setting.

day

Indicates that the probe summary statistics are to be cleared on one or more of the following days:

- ALL - every day
- MON - Monday
- TUE - Tuesday
- WED - Wednesday
- THU - Thursday
- FRI - Friday
- SAT - Saturday
- SUN - Sunday

Usage notes

- If you specify more than one day, separate the days with a blank space.
- Specify quotation marks before and after the days you specify.

Related statements

[IDS.ClearStat_Inform](#), [IDS.ClearStat_Log](#), [IDS.ClearStat_Log_File](#),
[IDS.ClearStat_Time](#)

IDS.ClearStat_Inform

Purpose

The IDS.ClearStat_Inform statement defines the inform policies that you use when probe summary statistics are cleared. Code as many of these statements as needed for your environment. The IDS.ClearStat_Inform statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.ClearStat_Inform statement has the following syntax:

►►IDS.—ClearStat.—Inform.—*suffix*—=—*inform_policy*————►►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each policy defined. In the example that follows, one inform policy (IDSOPERS) is defined:
IDS.ClearStat_Inform.1 = IDSOPERS

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.ClearStat_Inform.4 statement is ignored:

IDS.ClearStat_Inform.1
IDS.ClearStat_Inform.2
IDS.ClearStat_Inform.4

inform_policy

Specifies a predefined inform policy.

Usage notes

- The *inform_policy* must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined.
For information about defining inform policies, see Chapter 5, "Inform Policy Member," on page 493.

Related statements

IDS.ClearStat_Day, IDS.ClearStat_Log, IDS.ClearStat_Log_File, IDS.ClearStat_Time

IDS.ClearStat_Log

Purpose

The IDS.ClearStat_Log statement defines whether to log a report when the probe summary statistics are cleared. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.ClearStat_Log statement has the following syntax:

IDS.ClearStat_Log

►►[IDS.ClearStat_Log=N]
 |
 |IDS.ClearStat_Log=Y]————►►

where:

- N** Indicates that a report is not to be generated. This is the default setting.
- Y** Indicates that a report is to be generated when the probe summary statistics are cleared.

Usage notes

- The report is written to the DSIPARM data set member defined by the IDS.ClearStat_Log_File statement.

Related statements

IDS.ClearStat_Day, IDS.ClearStat_Inform, IDS.ClearStat_Log_File,
IDS.ClearStat_Time

IDS.ClearStat_Log_File

Purpose

The IDS.ClearStat_Log_File statement defines the DSIPARM data set member to which IDS probe summary statistics are written when the log is cleared. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.ClearStat_Log_File statement has the following syntax:

IDS.ClearStat_Log_File

►—IDS.ClearStat_Log_File—=—*member_name*—►

where:

member_name

Defines the DSIPARM data set member name. The initial setting is FKXSTATS.

Related statements

IDS.ClearStat_Day, IDS.ClearStat_Inform, IDS.ClearStat_Log, IDS.ClearStat_Time

IDS.ClearStat_Time

Purpose

The IDS.ClearStat_Time statement defines when to clear probe summary statistics. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.ClearStat_Time statement has the following syntax:

IDS.ClearStat_Time

►—IDS.ClearStat_Time—=—*hh:mm*—►

where:

hh:mm

Indicates the time of day in hours (24 hour clock) and minutes when the statistics are to be cleared. The initial default value is 00:00, indicating that the probe summary statistics are to be cleared at midnight.

Related statements

IDS.ClearStat_Inform, IDS.ClearStat_Log, IDS.ClearStat_Log_File,
IDS.ClearStat_Day

IDS.CONSOLEMSG

Purpose

The IDS.CONSOLEMSG statement specifies whether system console messages are used as the source of events for IDS automation services. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.CONSOLEMSG statement has the following syntax:

IDS.CONSOLEMSG

►►IDS.CONSOLEMSG=Y/N►►

where:

- Y Selects system console messages as the source of events.
- N Specifies that system console messages are not used as the source of events.

Usage notes

- For IDS automation services to be active, you must set either IDS.CONSOLEMSG or IDS.SYSLOGMSG to Y.
- For performance reasons, do not set both IDS.CONSOLEMSG and IDS.SYSLOGMSG to Y.

Note: If you specify both IDS.CONSOLEMSG=Y and IDS.SYSLOGMSG=Y, neither system console messages nor system log messages are used.

Related statements

IDS.SYSLOGMSG

IDS.DSIPARM

Purpose

The IDS.DSIPARM statement defines the DSIPARM data set to which all TCP/IP Intrusion Detection Services (IDS) reports are written. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the

NetView program to implement a change.

Syntax

The IDS.DSIPARM statement has the following syntax:

IDS.DSIPARM

►—IDS.DSIPARM—=—*data_set*—————►

where:

data_set

Defines the DSIPARM data set name. The IDSAUTO task must have write access to this data set.

IDS.Event_Inform

Purpose

The IDS.Event_Inform statement defines the inform policies that you use when an IDS event occurs. Code as many of these statements as needed for your environment. The IDS.Event_Inform statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Event_Inform statement has the following syntax:

►—IDS.—Event.—Inform.—*suffix*—=—*inform_policy*—————►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each policy used. In the example that follows, one inform policy (IDSOPERS) is defined:

IDS.Event_Inform.1 = IDSOPERS

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Event_Inform.4 statement is ignored:

IDS.Event_Inform.1
IDS.Event_Inform.2
IDS.Event_Inform.4

inform_policy

Specifies a predefined inform policy.

Usage notes

- The *inform_policy* must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined.

For information about defining inform policies, see Chapter 5, “Inform Policy Member,” on page 493.

IDS.Event_Limit

Purpose

The IDS.Event_Limit statement defines the number of event reports to keep. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Event_Limit statement has the following syntax:

IDS.Event_Limit

►►IDS.Event_Limit---nn--►►

where:

nn Specifies the number of event reports to keep. The number must be in the range of 1 - 99. The initial setting is 99.

Related statements

IDS.Event_Log, IDS.Event_Log_File

IDS.Event_Log

Purpose

The IDS.Event_Log statement defines whether to log an event and associated commands and responses to a DSIPARM data set. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Event_Log statement has the following syntax:

IDS.Event_Log

►► [IDS.Event_Log=N]
 |
 IDS.Event_Log=Y ►►

where:

- N** Indicates not to log the event and associated commands and responses. This is the default setting.
- Y** Indicates to log the event, commands, and responses to the DSIPARM data set defined by the IDS.Event_Log_File statement.

Related Statements

IDS.Event_Log_File, IDS.Event_Limit

IDS.Event_Log_File

Purpose

The IDS.Event_Log_File statement defines the DSIPARM data set member to which IDS events and their associated commands and responses are written. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Event_Log_File statement has the following syntax:

IDS.Event_Log_File

►—IDS.Event_Log_File—=—*member_name*—►

where:

member_name

Defines the DSIPARM data set member name. The name must be in the range of 1 - 6 characters. The initial setting is FKXIDS.

Usage notes

- If event reporting is enabled, each report is logged to a DSIPARM member named FKXIDS*nn* (or to the member name specified in the IDS.Event_Log_File statement), where *nn* is determined by the setting of the IDS.Event_Limit statement.

Related statements

IDS.Event_Log, IDS.Event_Limit

IDS.Flood_Cmd

Purpose

The IDS.Flood_Cmd statement specifies the command to be issued when an IDS attack detection flood event occurs. Code as many of these statements as needed for your environment. The IDS.Flood_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Flood_Cmd statement has the following syntax:

►—IDS.—Flood.—Cmd.—*suffix*—=—*command*—►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, one command is defined.

```
IDS.Flood_CmdType.1 = UNIX  
IDS.Flood_Cmd.1 = /bin/trmdstat -F -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Flood_Cmd.4 statement is ignored:

```
IDS.Flood_Cmd.1  
IDS.Flood_Cmd.2  
IDS.Flood_Cmd.4
```

command

Specifies the command to be issued.

Usage notes

- The IDS.Flood_CmdType statement specifies the command environment.

Related Statements

IDS.Flood_CmdType

IDS.Flood_CmdType

Purpose

The IDS.Flood_CmdType statement defines the command environment that you use when an IDS attack detection flood event occurs. Code as many of these statements as needed for your environment. The IDS.Flood_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Flood_CmdType statement has the following syntax:

```
►—IDS.—Flood_—CmdType.—suffix—=—MVS—  
                                                  |  
                                                 NETV  
                                                  |  
                                                 UNIX
```

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, only one command environment (UNIX) is defined.

```
IDS.Flood_CmdType.1 = UNIX  
IDS.Flood_Cmd.1 = /bin/trmdstat -F -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Flood_CmdType.4 statement is ignored:

```
IDS.Flood_CmdType.1  
IDS.Flood_CmdType.2  
IDS.Flood_CmdType.4
```

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command.

Usage notes

- Attack detection flood events have a probeid that begins with 0407.
- The IDS.Flood_Cmd statements define the commands to be issued.

Related statements

IDS.Flood_Cmd

IDS.probeid

Purpose

The IDS.probeid statement defines the supported IDS probeids. Code as many of these statements as needed for your environment. The IDS.probeid statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.probeid statement has the following syntax:

►—IDS.probeid.—suffix—=—probeid—►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each probeid defined.

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.probeid.4 statement is ignored:

```
IDS.probeid.1  
IDS.probeid.2  
IDS.probeid.4
```

probeid

Specifies the hexadecimal value of the probeid.

Usage notes

- To ignore a probe ID, add NONE to the end of the *probeid*, without blank characters. An example follows:

```
IDS.probeid.1 = 01002200(NONE)
```

Do not delete the *probeid*, otherwise you must renumber the suffix values to keep them all contiguous.

IDS.Report_Cmd

Purpose

The IDS.Report_Cmd statement specifies the command to be issued when the probe statistics report is generated. Code as many of these statements as needed for your environment. The IDS.Report_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Report_Cmd statement has the following syntax:

```
►—IDS.—Report.—Cmd_—suffix—=—command—————►
```

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, four commands are defined.

```
*  
IDS.Report_CmdType.1 = UNIX  
IDS.Report_Cmd.1 = /bin/trmdstat -I IDS.SYSLOG.FILENAME  
*  
IDS.Report_CmdType.2 = UNIX  
IDS.Report_Cmd.2 = /bin/trmdstat -A -S IDS.SYSLOG.FILENAME  
*  
IDS.Report_CmdType.3 = UNIX  
IDS.Report_Cmd.3 = /bin/trmdstat -U -S IDS.SYSLOG.FILENAME  
*  
IDS.Report_CmdType.4 = UNIX  
IDS.Report_Cmd.4 = /bin/trmdstat -T -S IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command

Specifies the command to be issued.

Usage notes

- The IDS.Report_CmdType statement specifies the command environment.

Related statements

IDS.Report_CmdType, IDS.Report_Inform, IDS.Report_Log, IDS.Report_Log_File

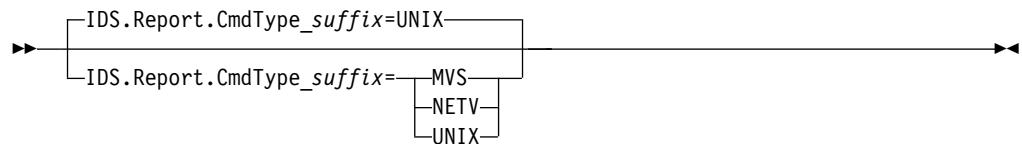
IDS.Report_CmdType

Purpose

The IDS.Report_CmdType statement defines the command environment that you use when the probe statistics report is generated. Code as many of these statements as needed for your environment. The IDS.Report_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Report_CmdType statement has the following syntax:



where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, one command environment (UNIX) is defined.

```
IDS.Report_CmdType.1 = UNIX
IDS.Report_Cmd.1 = /bin/trmdstat -I IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Report_CmdType.4 statement is ignored:

```
IDS.Report_CmdType.1
IDS.Report_CmdType.2
IDS.Report_CmdType.4
```

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage notes

- The IDS.Report_Cmd statements define the commands to be issued.

Related statements

IDS.Report_Cmd, IDS.Report_Inform, IDS.Report_Log, IDS.Report_Log_File

IDS.Report_Inform

Purpose

The IDS.Report_Inform statement defines the inform policies that you use when the probe statistics report is generated. Code as many of these statements as needed for your environment. The IDS.Report_Inform statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Report_Inform statement has the following syntax:

►—IDS.—Report __Inform.—*suffix*—=—*inform_policy*————►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each policy used. In the example that follows, one inform policy (IDSOPERS) is defined:

IDS.Report_Inform.1 = IDSOPERS

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Report_Inform.4 statement is ignored:

IDS.Report_Inform.1
IDS.Report_Inform.2
IDS.Report_Inform.4

inform_policy

Specifies a predefined inform policy.

Usage notes

- The *inform_policy* must be predefined in sample EZLINSMP. No checking is performed to verify that this policy is defined. For information about defining inform policies, see Chapter 5, “Inform Policy Member,” on page 493.

Related statements

IDS.Report_Cmd, IDS.Report_CmdType, IDS.Report_Log, IDS.Report_Log_File

IDS.Report_Log

Purpose

The IDS.Report_Log statement defines whether to log a timer-generated IDS probe statistics report to the DSIPARM data set member defined by the IDS.Report_Log_File statement. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Report_Log statement has the following syntax:

IDS.Report_Log



where:

N Indicates not to log the report. This is the default setting.

Y Indicates to log the report to the DSIPARM data set.

Related statements

IDS.Report_Cmd, IDS.Report_CmdType, IDS.Report_Inform, IDS.Report_Log_File

IDS.Report_Log_File

Purpose

The IDS.Report_Log_File statement defines the DSIPARM data set to which the IDS probe statistics reports are written. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Report_Log_File statement has the following syntax:

IDS.Report_Log_File



where:

member_name

Defines the DSIPARM data set member name. The initial setting is FNXREP.

Related statements

IDS.Report_Cmd, IDS.Report_CmdType, IDS.Report_Inform, IDS.Report_Log

IDS.Scan_Cmd

Purpose

The IDS.Scan_Cmd statement specifies the command to be issued when an IDS scan detection event occurs. Code as many of these statements as needed for your environment. The IDS.Scan_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for*

z/OS Installation: Getting Started. Stop and restart the NetView program to implement a change.

Syntax

The IDS.Scan_Cmd statement has the following syntax:

►►IDS.—Scan—Cmd.—suffix—=—command————►►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, one command is defined.

```
IDS.Scan_CmdType.1 = UNIX  
IDS.Scan_Cmd.1 = /bin/trmdstat -N -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Scan_Cmd.4 statement is ignored:

```
IDS.Scan_Cmd.1  
IDS.Scan_Cmd.2  
IDS.Scan_Cmd.4
```

command

Specifies the command to be issued.

Usage notes

- The IDS.Scan_CmdType statement specifies the command environment.

Related statements

IDS.Scan_CmdType

IDS.Scan_CmdType

Purpose

The IDS.Scan_CmdType statement defines the command environments that you use when an IDS scan detection event occurs. Code as many of these statements as needed for your environment. The IDS.Scan_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started.* Stop and restart the NetView program to implement a change.

Syntax

The IDS.Scan_CmdType statement has the following syntax:

►► [IDS.Scan_CmdType.suffix=UNIX]
[IDS.Scan_CmdType.suffix=
 | MVS
 | NETV
 | UNIX]————►►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, one command is defined.

```
IDS.Scan_CmdType.1 = UNIX  
IDS.Scan_Cmd.1 = /bin/trmdstat -N -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored. For example, if you define the following statements, the IDS.Scan_CmdType.4 statement is ignored:

```
IDS.Scan_CmdType.1  
IDS.Scan_CmdType.2  
IDS.Scan_CmdType.4
```

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage notes

- The IDS.Scan_Cmd statement defines the command to be issued.

Related statements

IDS.Scan_Cmd

IDS.SYSLOG.FILENAME

Purpose

The IDS.SYSLOG.FILENAME statement defines the UNIX System Services file to which IDS automation services are listening. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Note: Specify the IDS.SYSLOG.FILENAME statement only if IDS.SYSLOGMSG=Y.

Syntax

The IDS.SYSLOG.FILENAME statement has the following syntax:

IDS.SYSLOG.FILENAME

►►—IDS.SYSLOG.FILENAME—=—*filename*—►►

where:

filename

Defines the fully qualified UNIX System Services file name.

You can use the following shell variables in the file name:

Table 2. Supported Variables

Shell Variable	Definition	Substitution
%D	Day	<i>dd</i>
%M	Month	<i>mm</i>
%Y	Year	<i>yyyy</i>

Usage Note

Synchronize changes to the UNIX System Services syslog file, including clearing it, with the NetView program. The UNIX System Services syslog is updated using the **CRON** command. Recycle the NetView program after updates are made to the syslog file name.

IDS.SYSLOGMSG

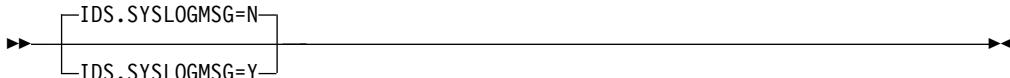
Purpose

The IDS.SYSLOGMSG statement defines the system log as the source of events for IDS automation services. This statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.SYSLOGMSG statement has the following syntax:

IDS.SYSLOGMSG



where:

- Y** Selects the system log as the source of events.
- N** Specifies not to use the system log as the source of events. This is the default setting.

Usage notes

- For IDS automation services to be active, you must set either IDS.CONSOLEMSG or IDS.SYSLOGMSG to Y.
- For performance reasons, do not set both IDS.CONSOLEMSG and IDS.SYSLOGMSG to Y.

Note: If you specify both IDS.CONSOLEMSG=Y and IDS.SYSLOGMSG=Y, neither system console messages nor system log messages are used.

Related Statements

IDS.CONSOLEMSG

IDS.TCP_Cmd

Purpose

The IDS.TCP_Cmd statement specifies the command to be issued when an IDS TCP traffic regulation event occurs. Code as many of these statements as needed for your environment. The IDS.TCP_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.TCP_Cmd statement has the following syntax:

►—IDS.—TCP—Cmd.—suffix—=—command—————►

where:

suffix

A numerical suffix. Increment the remaining suffixes by one for each command used. In the example that follows, one command is defined.

```
IDS.TCP_CmdType.1 = UNIX  
IDS.TCP_Cmd.1 = /bin/trmdstat -T -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command

Specifies the command to be issued.

Usage notes

- The IDS.TCP_CmdType statement specifies the command environment.

Related Statements

IDS.TCP_CmdType

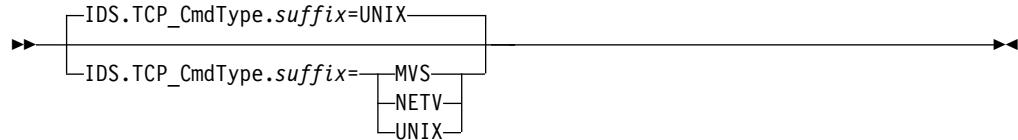
IDS.TCP_CmdType

Purpose

The IDS.TCP_CmdType statement defines the command environment that is used when an IDS TCP traffic regulation event occurs. Code as many of these statements as needed for your environment. The IDS.TCP_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.TCP_CmdType statement has the following syntax:



where:

suffix

A numeric suffix. Increment the remaining suffixes by one for each command environment that is specified. In the example that follows, one command environment (UNIX) is defined.

```
IDS.TCP_CmdType.1 = UNIX
IDS.TCP_Cmd.1 = /bin/trmdstat -T -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage notes

- The IDS.TCP_Cmd statements define the commands to be issued.

Related Statements

IDS.TCP_Cmd

IDS.UDP_Cmd

Purpose

The IDS.UDP_Cmd statement specifies the command to be issued when an IDS UDP traffic regulation event occurs. Code as many of these statements as needed for your environment. The IDS.UDP_Cmd statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.UDP_Cmd statement has the following syntax:

```
►—IDS.—UDP.—Cmd.—suffix—=—command————►
```

where:

suffix

A numeric suffix. Increment the remaining suffixes by one for each command defined. In the example that follows, one command is defined.

```
IDS.UDP_CmdType.1 = UNIX  
IDS.UDP_Cmd.1 = /bin/trmdstat -U -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

command

Specifies the command to be issued.

Usage notes

- The IDS.UDP_CmdType statement specifies the command environment.

Related statements

IDS.UDP_CmdType

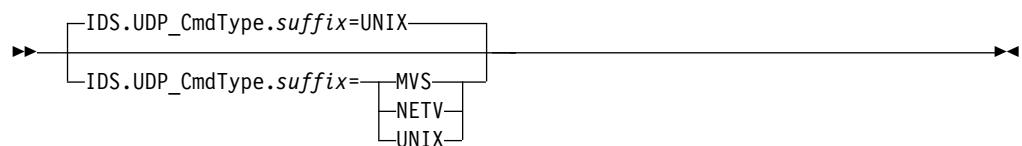
IDS.UDP_CmdType

Purpose

The IDS.UDP_CmdType statement defines the command environment that is used when an IDS UDP traffic regulation event occurs. Code as many of these statements as needed for your environment. The IDS.UDP_CmdType statement is located in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. Stop and restart the NetView program to implement a change.

Syntax

The IDS.UDP_CmdType statement has the following syntax:



where:

suffix

A numeric suffix. Increment the remaining suffixes by one for each command environment used. In the example that follows, one command environment (UNIX) is defined.

```
IDS.UDP_CmdType.1 = UNIX  
IDS.UDP_Cmd.1 = /bin/trmdstat -U -D IDS.SYSLOG.FILENAME
```

Note: Do not leave gaps in the suffix numbers because this gap causes statements with a higher sequence number to be ignored.

MVS

Indicates to issue an MVS command.

NETV

Indicates to issue a NetView command or command list.

UNIX

Indicates to issue a command to the UNIX System Services using the PIPE UNIX command. This is the default setting.

Usage notes

- The IDS.UPD_Cmd statements define the commands to be issued.

Related statements

IDS.UPD_Cmd

INFORM.POLICY.MEMBER

Purpose

Specifies the DSIPARM data set member that contains the INFORM policy that is loaded during initialization.

Syntax

The INFORM.POLICY.MEMBER statement has the following syntax:

INFORM.POLICY.MEMBER

►►—INFORM.POLICY.MEMBER=*member_name*—————►►

where:

member_name

Specifies the DSIPARM data set member that contains the INFORM policy.

Usage note

The INFORM.POLICY.MEMBER statement overrides any INFORMPM parameters that might be specified on any policy ENVIRON SETUP statement.

INIT.DVIPASTATS

Purpose

The INIT.DVIPASTATS statement indicates whether to enable logging for distributed DVIPA statistics at NetView initialization. Logging occurs when data is discovered by the distributed DVIPA discovery function.

Syntax

The INIT.DVIPASTATS statement has the following syntax:

INIT.DVIPASTATS

►►—

INIT.DVIPASTATS = NO
INIT.DVIPASTATS = YES

—————►►

where:

NO Indicates not to log distributed DVIPA statistics

YES

Indicates to log distributed DVIPA statistics.

Usage notes

- The INIT.DVIPASTATS statement requires that the DVIPA.DVTAD subtower (under the DVIPA tower) be enabled.
- The sequential data sets to which distributed DVIPA statistics are written are allocated by the CNMSJ002 sample. The CNMDVIPP and CNMDVIPS DD statements in the NetView startup procedure CNMPROC (CNMSJ009) refer to these data sets.
- When the INIT.DVIPASTATS statement is set to YES and the DVIPALOG LIST command is run on a system where a backup distributed DVIPA is not yet active, the following information is displayed as part of the command output:
DVIPALOG.ActiveLog = Undetermined

Note that even though this information is displayed, data is logged by the NetView program on the system that owns the distributed DVIPA.

INIT.EMAAUTO

Purpose

The INIT.EMAAUTO statement specifies whether the EMAAUTO.TEMS.procStr statement or the EMAAUTO.TEMA.procStr is to be processed.

Syntax

The INIT.EMAAUTO statement has the following syntax:

INIT.EMAAUTO



where:

NO Indicates not to process either the EMAAUTO.TEMS.procStr or the EMAAUTO.TEMA.procStr statement.

YES

Indicates to process the EMAAUTO.TEMS.procStr statement and the EMAAUTO.TEMA.procStr statement.

Usage notes

- The INIT.EMAAUTO statement requires that the TEMA tower be enabled.

INIT.NRM

Purpose

The INIT.NRM statement starts the NetView Resource Manager.

Syntax

The INIT.NRM statement has the following syntax:

INIT.NRM



where:

NO Indicates not to start the NetView Resource Manager.

YES

Indicates to start the NetView Resource Manager.

Usage notes

- You can start the NetView Resource Manager at a later time by issuing the INITNRM command.
- If you run TOWER = GRAPHICS and use the NetView Resource Manager, set the INIT.NRM statement to YES.
- If the INIT.NRM statement is set to YES and the NRM.TYPE statement is set to AGENT, code at least one NRM.HOSTDEST statement and the NRM.CMODE and NRM.PORT statements as required for your system.

INIT.OPKT

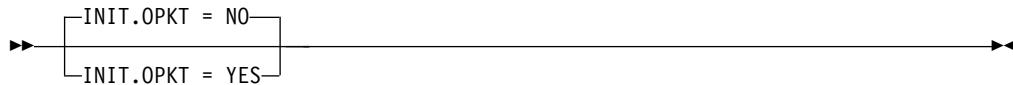
Purpose

The INIT.OPKT statement specifies whether to start the OSA packet trace management function at NetView initialization. If this function is enabled, the PKTS DEFINE and PKTS START commands are issued for each defined stack/autotask combination.

Syntax

The INIT.OPKT statement has the following syntax:

INIT.OPKT



where:

NO Indicates not to start the OSA packet trace management function.

YES

Indicates to start the OSA packet trace management function.

Usage notes

- The INIT.OPKT statement requires that the TCPIPCOLLECT.PKTS subtower be enabled.

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

INIT.PKTS

Purpose

The INIT.PKTS statement specifies whether to start the IP packet trace management function at NetView initialization. If this function is enabled, the PKTS DEFINE and PKTS START commands are issued for each defined stack/autotask combination.

Syntax

The INIT.PKTS statement has the following syntax:

INIT.PKTS



where:

NO Indicates not to start the IP packet trace management function.

YES

Indicates to start the IP packet trace management function.

Usage notes

- The INIT.PKTS statement requires that the TCPIPCOLLECT.PKTS subtower be enabled.
- You can start IP packet trace management at a later time by issuing the PKTS START command.

INIT.TCPCONN

Purpose

The INIT.TCPCONN statement specifies whether to start TCP/IP connection management at NetView initialization.

Syntax

The INIT.TCPCONN statement has the following syntax:

INIT.TCPCONN



where:

NO Indicates not to start TCP/IP connection management.

YES

Indicates to start TCP/IP connection management.

Usage notes

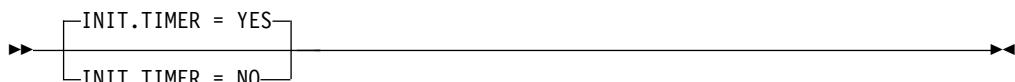
- The INIT.TCPCONN statement requires that the TCPIPCOLLECT.TCPCONN subtower be enabled.
- Specifying the INIT.TCPCONN statement to YES causes the TCPConn DEFINE and TCPConn START commands to be issued.
- You can start TCP/IP connection management at a later time by issuing the TCPConn START command.

INIT.TIMER**Purpose**

The INIT.TIMER statement specifies whether to issue the RESTORE TIMER command during NetView initialization.

Syntax

The INIT.TIMER statement has the following syntax:

INIT.TIMER

where:

NO Indicates not to issue the RESTORE TIMER command.

YES

Indicates to issue the RESTORE TIMER command.

Usage notes

- If you set the INIT.TIMER statement to NO, the CATCHUP function in the AON component is not started. You can restore timer command data at a later time by issuing the RESTORE TIMER command. For more information, refer to the *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)* or the NetView online help.

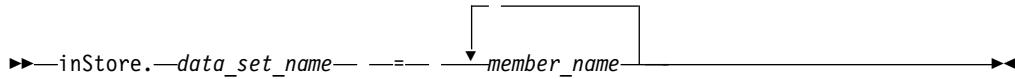
inStore**Purpose**

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

Syntax

The inStore statement has the following syntax:

inStore



where:

data_set_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 *IBM Tivoli NetView for z/OS Programming: Pipes* or the NetView online help.

IPLOG

Purpose

The IPLOG statement defines TCP/IP definitions for the syslog server (task DSIIIPLOG, member DSILGCF).

Syntax

The IPLOG statement has the following syntax:

IPLOG



where:

TCPANAME = name

Defines the TCP/IP address space name.

PORT = port

Defines the port number on which DSIIIPLOG waits for connection requests. The default is 514.

SOCKETS = number

Specifies the maximum number of simultaneous users. The default is 100.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE IPLOG command.

IPv6Env

Purpose

The IPv6Env statement defines IPv6 networking preferences for the current NetView domain.

Syntax

The IPv6Env statement has the following syntax:

IPv6Env



where:

MIXED

NetView supports both IPv4 and IPv6 networking. In general, if there is a choice, the NetView program uses IPv6 addressing. If an IP address is explicitly specified, the NetView program uses the appropriate IP transport:

- IPv4 (INET) for IPv4 addresses
- IPv6 (INET6) for IPv6 addresses

It is possible to process both IPv4 and IPv6 addresses at the same time (for example, listening on an AFINET6 socket for both IPv4 datagrams and IPv6 datagrams, either of which can represent SNMP traps).

ONLY

The NetView program generally uses IPv6 notation and transports, even if IPv4 addresses are explicitly specified.

IPv4 host names cannot be used when the IPv6ENV statement is set to ONLY.

NONE

The NetView program uses only IPv4 transports, even when a valid IPv6 address is specified.

Attention: An IPv4-compatible IPv6 address might be recognized as a valid IPv4 address and handled accordingly.

iverbose

Purpose

The iverbose statement is used to specify whether NetView initialization messages and command echoes that are generated by CNMSTYLE processing are suppressed

from the system console and the system log. These messages are still sent to the NetView log.

Syntax

The iverbose statement has the following syntax:

iverbose



where:

- A** Specifies that no suppression is performed. This is the default value if the iverbose statement is not coded in the CNMSTYLE member.
- W** Specifies that many NetView initialization messages associated with CNMSTYLE processing are suppressed. Commands are echoed to the NetView console.

Usage notes

- The DSI112I message indicates the end of base NetView initialization.

JesJobLog

Purpose

Use the JesJobLog statement to allocate the JES job log.

Syntax

The JesJobLog statement has the following syntax:

JesJobLog



where:

- YES** Specifies to allocate the JES job log. This is the default.
- NO** Specifies that the JES job log is not to be allocated.

LOADEXIT

Purpose

Use the LOADEXIT statement to specify the installation exits that you want to load. The NetView program attempts to load only the exits that are specified. Code one LOADEXIT statement for each installation exit to be loaded.

Syntax

The LOADEXIT statement has the following syntax:

LOADEXIT



where:

exit_name

Indicates one of the following installation exits:

- DSIE01
- DSIE02A
- DSIE03
- DSIE04
- DSIE05
- DSIE06
- DSIE07
- DSIE09
- DSIE10
- DSIE11
- DSIE12
- DSIE13
- DSIE14
- DSIE16
- DSIE16B
- DSIE17
- DSIE18
- DSIE19
- DSIE20
- DSIE21

YES

Specifies to load the exit.

NO

Specifies not to load the exit. This is the default.

Usage notes

- DSIE15 is an obsolete exit, but is supported if used in your environment.
- Refer to the *IBM Tivoli NetView for z/OS Programming: Assembler* for information about installation exits.

LOGONPW

Purpose

Use the LOGONPW statement to specify whether the NetView logon screen can be bypassed by operators specifying their password as part of the VTAM logon specification.

Syntax

The LOGONPW statement has the following syntax:



where:

NO Specifies that operators cannot bypass the NetView logon screen.

YES

Specifies that operators can bypass the NetView logon screen.

LUC

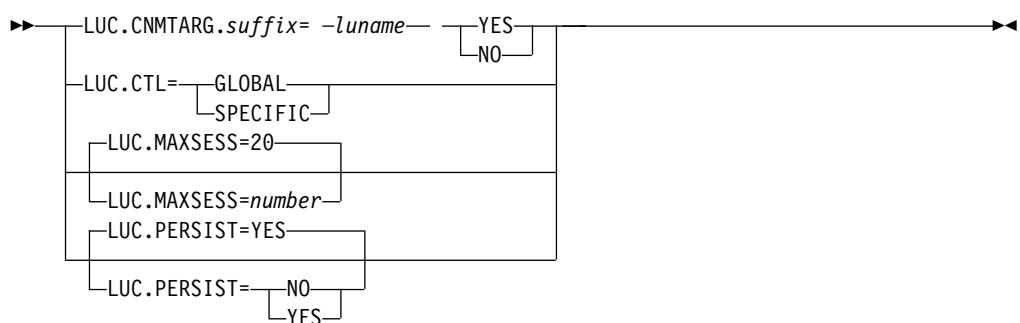
Purpose

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

Syntax

The CNM statement has the following syntax:

LUC



where:

LUC.CNMTARG.suffix = *lname* [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 *xxxxxLUC*, where *xxxxx* 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 *IBM 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 that no authority check is done when the NetView program establishes new NNT cross-domain session monitor sessions. The system accepts any session initiation request. This is the initial setting.

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 *IBM 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 are to 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 LU.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 definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE LUC command.

Related statements

CDRMDEF, DOMAINS, LU, TASK.

memStore

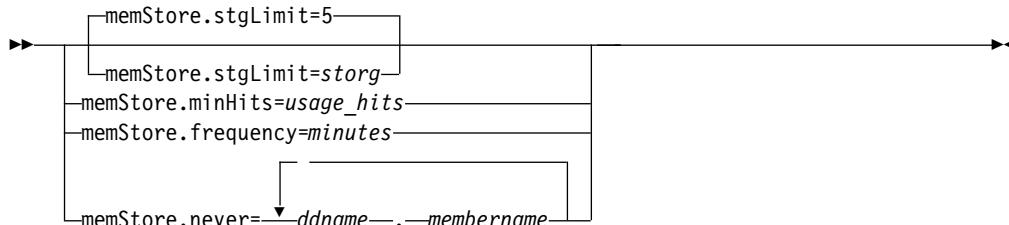
Purpose

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 processor usage when accessing PDS members from the NetView program.

Syntax

The memStore statement has the following syntax:

memStore



where:

stgL=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 fewer than this is not loaded into storage.

frequency=minutes

Specifies the time interval in minutes that memStore tests for usage.

never=ddname.membername

Specifies the *ddname* and the associated *membername* that memStore does not cache. If *ddname* is not specified, or an asterisk (*) is specified for *ddname*, the action is against the specified *membername* in all DD data sets defined to the

NetView program. If an asterisk is specified for *membername*, the action is against all member names for the specified *ddname*. Use a period to separate *ddname* from *membername*.

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 the CNMSTYLE member and to the MEMSTORE command in the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE MEMSTORE command.

MODIFY.TOWER

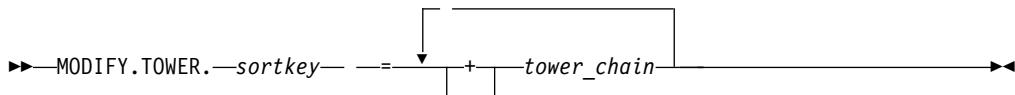
Purpose

The MODIFY.TOWER statement enables or disables the specified towers and subtowers. Enablement or disablement is considered for only the final tower or subtower name that is specified in the tower chain.

Syntax

The MODIFY.TOWER statement has the following syntax:

MODIFY.TOWER



where:

sortkey

The sort key determines the order of processing of MODIFY.TOWER statements. The sort key must be alphanumeric and limited to eight characters. If statements with identical sort keys are discovered, only the last identical statement is considered; all others are ignored. The remaining statements are then sorted alphabetically, with the letter A coming before the numeral zero (0).

tower_chain

Specifies a sequence of tower and subtower names, where each name is separated by a period. Each name can be up to 16 characters long. Each sequence must be preceded by a plus (+) sign or minus (-) sign. A plus sign enables the sequence; a minus sign disables the sequence. Tower and subtower names can only include alphabetic, numeric, or national characters (@#\$).

Usage notes

- The MODIFY.TOWER statement adds or deletes towers and subtowers. You can specify as many MODIFY.TOWER statements as you require.
- MODIFY.TOWER statements are processed immediately after all TOWER statements are processed, in the alphabetical order specified by the *sortkey*.

- Enablement of a subtower is conditional; the request is considered for only the final name specified. The tower statement or subtower prior to the final name is first evaluated, and enablement continues only if the tested tower statement or subtower is enabled.
- Before using the MODIFY.TOWER statement, refer to “TOWER” on page 263.
- Enablement or disablement of the *tower_chain* sequence can be tested using the REXX tower() function. Enabling or disabling tower chains affects the availability and function of various components of the NetView program and of some supported applications, such as System Automation for z/OS.
- Tower and subtower names are not case sensitive.

Examples

1. Consider the following example MODIFY.TOWER statement, which includes two *tower.subtower* sequences:

```
MODIFY.TOWER.Aa = +AON.SNA -Graphics.SNATM
```

In this example, the sort order for the statement is interpreted as AA, and is therefore evaluated early on in processing.

The first request is to enable the sequence AON.SNA. First, tower AON is evaluated. In this example, AON is disabled, so nothing is done. If a subsequent statement enables tower AON, that does not imply that AON.SNA would inherit any status from this action.

Note that the sequence +AON +AON.SNA would enable both tower AON and the subtower SNA.

The next request to disable the sequence Graphics.SNATM. If tower Graphics is disabled, nothing is done. If subtower SNATM is disabled, nothing is done. However, if prior processing has enabled both Graphics and SNATM, the result is that Graphics.SNATM is disabled.

2. Consider the following example *tower.subtower* sequence:

```
+Andy.Brenda.Charles.Donna
```

The first step checks for whether Andy.Brenda.Charles is enabled. If it is disabled, then nothing further is done. The next step checks for whether Andy.Brenda.Charles.Donna is enabled. If it is enabled, then nothing further is done and no message is issued. If it is disabled, then the name Donna is enabled as a subtower of Andy.Brenda.Charles.

To enable the entire sequence, specify the four name sequences in order. For example:

```
MODIFY.TOWER.xyz = +Andy +andy.Brenda +andy.brenda.Charles +andy.bRenda.Charles.donna
```

This logic also applies to the disablement of towers and subtowers. Only the final name is subject to action—if any previous towers or subtowers are found to be disabled, the entire sequence, as a value of a MODIFY.TOWER statement, is ignored.

MSG.TECROUTE

Purpose

The MSG.TECROUTE statement specifies the name of the PPI receiver that is associated with the Event/Automation Service. This statement is coded in the CNMSTIDS member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Syntax

The MSG.TECROUTE statement has the following syntax:



where:

name

The PPI receiver name. The default value is IHSATEC.

MVSPARM.ActionDescCodes

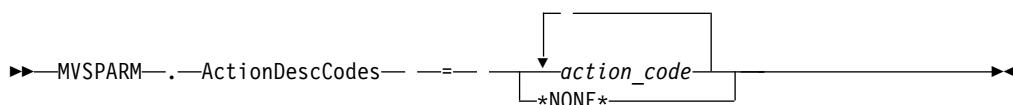
Purpose

The MVSPARM.ActionDescCodes statement specifies a list of action descriptor codes. Messages with these descriptor codes and WTORs are considered action messages by the NetView program.

Syntax

The MVSPARM.ActionDescCodes statement has the following syntax:

MVSPARM.ActionDescCodes



where:

action_code

Specifies the action code. The value can be from 1-16. You can specify multiple descriptor codes.

NONE

Specifies that only WTORs are considered action messages.

Usage notes

- By default, action messages have a small amount of storage that the NetView program uses to process a corresponding DOM. Storage for action messages for which a DOM is never received can build to the point that the NetView program runs out of storage, severely impacting operations. If you specify a value that is not a default value, consider whether you can be certain that all matching messages are deleted by a DOM in a timely manner. If not, you can control storage growth in the NetView program by using one of the following methods:
 - In your automation table, after automating any messages of particular interest, automate all other messages having the subject descriptor code and include a DOMACTION(NODELMSG) automation action.
 - Use NetView automation that acts on the MAXCSSIR DEFAULTS value. See the DSITBL01 sample for automation of the BNH535A message that triggers cleanup of stale action messages.

- If you do not include an MVSPARM.ActionDescCodes statement, WTORs and messages with descriptor codes 1, 2, and 11 are considered action messages.
- Messages that are received from MVS that contain one of the listed descriptor codes are marked as HELD and have a default DOMACTION of DELMSG.

MVSPARM.Cmd.Designator

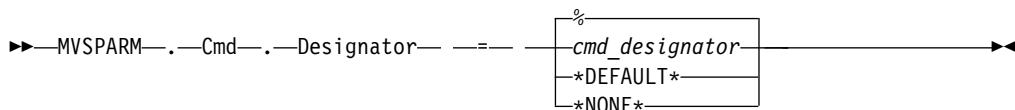
Purpose

The MVSPARM.Cmd.Designator statement specifies a unique designator character that is used to prefix NetView commands that are entered from a z/OS system console.

Syntax

The MVSPARM.Cmd.Designator statement has the following syntax:

MVSPARM.Cmd.Designator



where:

cmd_designator

Specifies the command designator character. The value can be 1 - 8 characters and must be printable and not blank.

DEFAULT

Indicates to use the 4-character NetView subsystem name.

NONE

Indicates that the NetView program is not to register or use any command designator.

Usage notes

- You do not have to have the NetView subsystem address space active to forward commands to the NetView program (using the designator character). You can also issue NetView commands from the z/OS system console by using the MVS MODIFY command or by using the NetView Command Revision function. For information about issuing commands from the z/OS system console, see the *IBM Tivoli NetView for z/OS Automation Guide*.
- You can use the MVSPARM.Cmd.Scope to specify the scope of the command designator (local system or sysplex).
- The command designator must precede all NetView commands and command lists that are issued from an MVS console to distinguish them from other z/OS commands. The default value (*DEFAULT*) causes the subsystem to use the 4-character subsystem name as the prefix.

If you are running two NetView programs in the same host, the subsystem start procedures for the second copy must specify different characters from the ones that are used for the first copy. The designator must also be different from the one that is used by JES or any other subsystem. If multiple subsystems in the

same host use the same command designator, commands from MVS consoles starting with that character are passed to each of the subsystems to be processed.

If you are running two NetView programs on the same system, refer to *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* in the Appendix entitled *Running Multiple NetView Programs in the same LPAR*.

MVSPARM.Cmd.Scope

Purpose

The MVSPARM.Cmd.Scope statement specifies the scope of the command designator that you specified with the MVSPARM.Cmd.Designator statement.

Syntax

The MVSPARM.Cmd.Scope statement has the following syntax:

MVSPARM.Cmd.Scope



where:

- 1 Specifies that the command designator can be used on a local system. This is the default value.
 - * Specifies that the command designator can be used for the sysplex.

MVSPARM.DEFAUTH

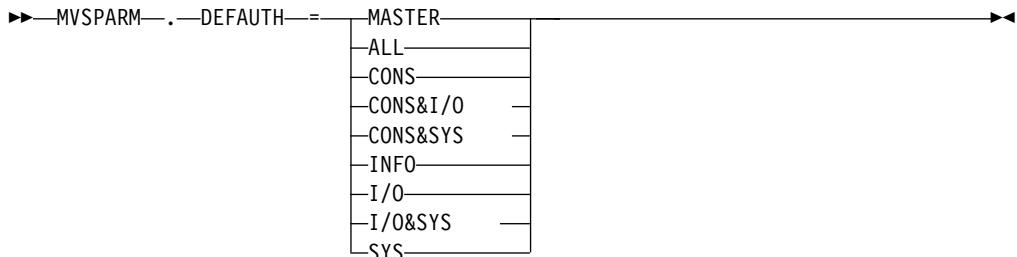
Purpose

The MVSPARM.DEFAUTH statement specifies the default authority for the EMCS consoles that are obtained by the NetView program.

Syntax

The MVSPARM.DEFAUTH statement has the following syntax:

MVSPARM.DEFAUTH



where:

MASTER

The EMCS console can enter all possible MVS commands. This is the default value.

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.Msg.Automation

Purpose

The MVSPARM.Msg.Automation statement specifies whether MVS messages are imported for automation.

Syntax

The MVSPARM.Msg.Automation statement has the following syntax:

MVSPARM.Msg.Automation



where:

Yes

Indicates that MVS messages that are marked for automation in the message processing facility (MPF) or the message revision table (MRT) are sent to message automation. Yes is the default value.

No Indicates that MVS messages are not submitted to message automation.

Specify No if another instance of the NetView program (in the same LPAR) will automate MVS messages, but you want to have the CNMCSSIR task active for its other functions.

Usage notes

- The MVSPARM.Msg.Automation statement setting does not affect responses to MVS commands or messages that are directed to a console obtained by the GETCONID command.

MVSPARM.Msg.Automation.MaxAge

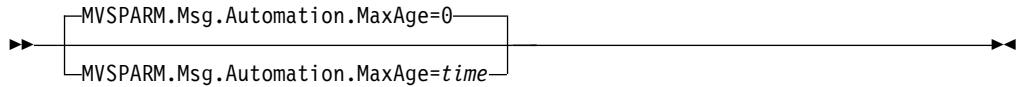
Purpose

The MVSPARM.Msg.Automation.MaxAge statement indicates the disposition of MVS messages that are marked for automation while the NetView program or the subsystem router task (CNMCSSIR) is inactive. If the age of a queued message exceeds the value of the MVSPARM.Msg.Automation.MaxAge statement, the message is not submitted for automation by the NetView program.

Syntax

The MVSPARM.Msg.Automation.MaxAge statement has the following syntax:

MVSPARM.Msg.Automation.MaxAge



where:

time

Specifies the time in seconds. The default value is 0.

Usage notes

- The MVSPARM.Msg.Automation.MaxAge statement is ignored if the MVSPARM.Msg.Automation statement is set to No.

Example

The NetView program is inactive. During this period, the following messages are issued:

- EXX111E at 10:30:05
- EXX222A at 10:30:12

The NetView program then becomes active at 10:30:15 (the MVSPARM.Msg.Automation.MaxAge statement is set to 5 seconds). The EXX222A message is made available to be automated by the NetView program. However, the EXX111E message is discarded because it is greater than 5 seconds old.

MVSPARM.Msg.Automation.Oldtag

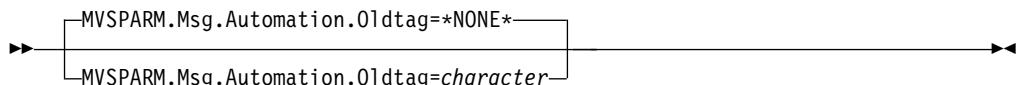
Purpose

When MVS messages are queued for later automation because the NetView program or the subsystem router task (CNMCSSIR) is inactive, the NetView program overlays the last (eighth) character of the autotoken field with the character that is specified on the MVSPARM.Msg.Automation.Oldtag statement.

Syntax

The MVSPARM.Msg.Automation.Oldtag statement has the following syntax:

MVSPARM.Msg.Automation.Oldtag



where:

character

Specifies a non-blank character.

NONE

Specifies not to overlay the last character. This is the default value.

Usage notes

- The MVSPARM.Msg.Automation.Oldtag statement is ignored if the MVSPARM.Msg.Automation.MaxAge statement is set to 0.

MVSPARM.OperRecvBrdcst

Purpose

The MVSPARM.OperRecvBrdcst statement specifies whether operator tasks that have acquired an MVS console receive broadcast messages from the MVS program.

Syntax

The MVSPARM.OperRecvBrdcst statement has the following syntax:

MVSPARM.OperRecvBrdcst



where:

Yes

All NetView operator tasks other than the PPT that have acquired an MVS console receive broadcast messages from the MVS program. This is the default value.

No No NetView operator tasks other than the CNMCSSIR task receive broadcast messages from the MVS program.

Usage notes

- The BRDCST parameter of the GETCONID command overrides the MVSPARM.OperRecvBrdcst value.
- If the EMCSPARM value is set to SAF using the DEFAULTS or OVERRIDE command, the broadcast message setting that is specified using the OPERPARM segment in the resource access control facility (RACF) or an equivalent security access facility (SAF) product overrides the MVSPARM.OperRecvBrdcst value and the GETCONID BRDCST value.
- To allow the PPT to receive broadcast messages, issue a RELCONID command (if necessary) and then a GETCONID command specifying BRDCST=RECV to the PPT.

NACMD.DESTPPI

Purpose

The NACMD.DESTPPI statement specifies the program-to-program interface (PPI) receiver name used by the NetView for z/OS Enterprise Management Agent.

Syntax

The NACMD.DESTPPI statement has the following syntax:

NACMD.DESTPPI



where:

ppi_name

Specifies the name of the PPI receiver. The default value is CNMEMARX.

For information about accepted values of the PPI receiver name, see the RECEIVERT-ID section in the *IBM Tivoli NetView for z/OS Application Programmer's Guide*. Additionally, the PPI receiver name must match the value specified during configuration of the NetView for z/OS Enterprise Management Agent. For more information, refer to the *IBM Tivoli NetView for z/OS Installation: Configuring the NetView Enterprise Management Agent*.

Usage notes

- The value for NACMD.DESTPPI is assigned in the following way:
 1. The DESTPPI value specified with the NACMD command
 2. The value of the CNMSTYLE.NACMD.DESTPPI variable
 3. The default value of CNMEMARX.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.INTCONINACT

Purpose

The NACMD.INTCONINACT statement defines the time interval for data collectors to collect inactive TCPIP connection data for the NetView for z/OS Enterprise Management Agent to use.

Syntax

The NACMD.INTCONINACT statement has the following syntax:

NACMD.INTCONINACT



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 3600 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.INTCONNACT

Purpose

The NACMD.INTCONNACT statement defines the time interval for data collectors to collect TCPIP connection data for the NetView for z/OS Enterprise Management Agent to use.

Syntax

The NACMD.INTCONNACT statement has the following syntax:

NACMD.INTCONNACT



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 900 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.INTHEALTH

Purpose

The NACMD.INTHEALTH statement defines the time interval for data collectors to collect NetView task data for the NetView for z/OS Enterprise Management Agent to use.

Syntax

The NACMD.INTHEALTH statement has the following syntax:

NACMD.INTHEALTH



where:

seconds

Specifies the time interval in the range 30 - 43200 seconds. The default value is 30 seconds.

Usage notes

- You might want to set the NACMD.INTHEALTH interval similar to the DISCOVERY.INTAPPL interval.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.INTSESSACT

Purpose

The NACMD.INTSESSACT statement defines the time interval for data collectors to collect session data for the NetView for z/OS Enterprise Management Agent to use.

Syntax

The NACMD.INTSESSACT statement has the following syntax:

NACMD.INTSESSACT



where:

seconds

Specifies the time interval in the range 60 - 43200 seconds. The default value is 900 seconds.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.LCLPPIRV

Purpose

The NACMD.LCLPPIRV statement specifies the name of the local PPI sender name that is used by the NetView for z/OS Enterprise Management Agent.

Syntax

The NACMD.LCLPPIRV statement has the following syntax:

NACMD.LCLPIRV

►—NACMD.LCLPIRV— = —*localPPI_sender_name*—►

where:

localPPI_sender_name

Specifies the name of the local PPI sender.

For information on accepted values of the PPI sender name, see the SENDER-ID section in the *IBM Tivoli NetView for z/OS Application Programmer's Guide*. Additionally, the PPI sender name must match the value specified during configuration of the NetView for z/OS Enterprise Management Agent. For more information, refer to the *IBM Tivoli NetView for z/OS Installation: Configuring the NetView Enterprise Management Agent*.

Usage notes

- The value for the local PPI sender is assigned in the following way:
 1. The LCLPIRV value specified with NACMD.
 2. The NACMD.LCLPIRV statement in the CNMSTYLE member.
 3. The default value of the NetView domain.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.OPID

Purpose

The NACMD.OPID statement specifies the Tivoli Enterprise Portal (TEP) logon ID that you want to map to a NetView operator ID.

Syntax

The NACMD.OPID statement has the following syntax:

NACMD.OPID

►—NACMD.OPID.—*TEPLogonid*— —=— —*netview_operator_id*—►

where:

TEPLogonid

Specifies the Tivoli Enterprise Portal logon ID.

netview_operator_id

Specifies the NetView operator ID.

Usage notes

- Tivoli Enterprise Portal user IDs can be greater than 8 characters in length. If they are sent to the NetView program, a Tivoli Enterprise Portal user ID can be mapped to a NetView operator ID.

Tivoli Enterprise Portal user IDs are sent to the NetView program if you use the following security features of the hub Tivoli Enterprise Monitoring Server or the NetView for z/OS Enterprise Management Agent:

- Password authentication by the hub Tivoli Enterprise Monitoring Server
- Command authorization for Take Action commands

For additional information on these security features, refer to *IBM Tivoli NetView for z/OS Installation: Configuring the NetView Enterprise Management Agent* or *IBM Tivoli NetView for z/OS Security Reference*.

- The *netview_operator_id* must follow the definition rules for a NetView operator ID. For information on the NetView operator ID, refer to *IBM Tivoli NetView for z/OS Security Reference*.
- The mapping fails if *TEPLogonid* contains characters that are not supported by common global variable names.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.PERSIST

Purpose

For debugging purposes, the NACMD.PERSIST statement is used to preserve the data space that is associated with the NetView for z/OS Enterprise Management Agent when NACMD processing ends.

Syntax

The NACMD.PERSIST statement has the following syntax:

NACMD.PERSIST

►—NACMD.PERSIST— = —*hours*—►

where:

hours

Specifies the amount of time in hours that the data space is kept active after the NACMD command ends. If a value of 0 is specified, the data space is deleted. The default value is 0.

Usage notes

- The value for PERSIST is assigned in the following way:
 1. The PERSIST value specified with the NACMD command.
 2. The value of the CNMSTYLE.NACMD.PERSIST variable.
 3. The default value is 0 and the data space is deleted when the NACMD command ends.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.ROWSxxx

Purpose

The NACMD.ROWSxxx statement defines the number of rows of data to collect for the Tivoli Enterprise Portal workspaces that are associated with the NetView for z/OS Enterprise Management Agent.

Syntax

The NACMD.ROWSxxx statement has the following syntax:

NACMD.ROWSxxx

►—NACMD.ROWSxxx=number—————►

where:

ROWSxxx

Specifies the number of rows of data to collect for each of the workspaces. The following values can be used:

ROWSALOG

NetView Audit Log workspace. The default number of rows is 50000. The maximum number of rows is 200000.

ROWSCONINACT

Inactive TCPIP Connection Data workspace. The default number of rows is 20000. The maximum number of rows is 80000.

ROWSCONNACT

TCPIP Connection Data workspace. The default number of rows is 200000. The maximum number of rows is 800000.

ROWSDVCONN

DVIPA Connections workspace. The default number of rows is 10000. The maximum number of rows is 40000.

ROWSDVDEFDS

DVIPA Definition and Status workspace. The default number of rows is 10000. The maximum number of rows is 40000.

ROWSDVROUTCR

Distributed DVIPA Connection Routing workspace. The default number of rows is 150000. The maximum number of rows is 600000.

ROWSDVROUTVR

VIPA Routes workspace. The default number of rows is 256. The maximum number of rows is 1024.

ROWSDVTADDT

Distributed DVIPA Targets workspace. The default number of rows is 10000. The maximum number of rows is 40000.

ROWSDVTADSD

DVIPA Sysplex Distributors workspace. The default number of rows is 10000. The maximum number of rows is 40000.

ROWSDVTADSH

Distributed DVIPA Server Health workspace. The default number of rows is 10000. The maximum number of rows is 40000.

ROWSHEALTH

NetView Tasks workspace. The default number of rows is 500. The maximum number of rows is 4096.

ROWSHIPERSOCK

HiperSockets Configuration and Status workspace. The default number of rows is 256. The maximum number of rows is 1024.

ROWSNVAPPS

NetView Applications workspace. The default number of rows is 256. The maximum number of rows is 1024.

ROWSNVCMD

NetView Command Response workspace. The default number of rows is 100000. The maximum number of rows is 400000.

ROWSNVLOG

NetView Log workspace. The default number of rows is 1000. The maximum number of rows is 4000.

ROWSOSA

OSA Channels and Ports workspace. The default number of rows is 128. The maximum number of rows is 256.

ROWSSESSACT

Session Data workspace. The default number of rows is 200000. The maximum number of rows is 800000.

ROWSSTCKCFG

Stack Configuration and Status workspace. The default number of rows is 256. The maximum number of rows is 1024.

ROWSTELCFG

Telnet Server Summary table view in the Telnet Server Configuration and Status workspace. The default number of rows is 256. The maximum number of rows is 1024.

ROWSTELPRT

Telnet Server Port Summary table view in the Telnet Server Configuration and Status workspace. The default number of rows is 256. The maximum number of rows is 1024.

number

The number of rows. The default and maximum values are listed with the workspace row name.

Usage notes

To implement definition changes for the NACMD.ROWS statement, the NetView program must be recycled. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NACMD.SUBNODE

Purpose

The NACMD.SUBNODE statement defines the subnode name to use for this NetView domain in the Tivoli Enterprise Portal under the NetView for z/OS Enterprise Management Agent. Use of this statement is optional. If omitted, the subnode name will default to the NetView domain name.

Syntax

The NACMD.SUBNODE statement has the following syntax:

NACMD.SUBNODE

►►—NACMD.SUBNODE=*nodename*————►►

where:

nodename

Specifies the subnode name to use for the domain in the Tivoli Enterprise Portal. The value can be up to 32 characters in length.

Usage notes

Subnode names can be 1-32 characters in length, consisting of alphanumeric characters and the special characters * . _ - : @ \$ #. A subnode name cannot start with the special characters * . #. Blank spaces are not supported in the subnode name.

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

The changes do not go into effect until a subsequent NACMD command is issued to establish a connection to the agent.

NACMD.WAITSECS

Purpose

The NACMD.WAITSECS statement defines how long to wait for a response from a Take Action command issued from the Tivoli Enterprise Portal using the NetView for z/OS Enterprise Management Agent.

Syntax

The NACMD.WAITSECS statement has the following syntax:

NACMD.WAITSECS

►►—NACMD.WAITSECS = *seconds*————►►

where:

seconds

Specifies the number of seconds to wait for a command response. If you specify 0 (no wait), the default value of 90 seconds is used. The valid value range is 1 - 10000000.

Usage notes

- The value for WAITSECS is assigned in the following way:
 1. The WAITSECS value specified with the NACMD command, if not null

2. The value of the CNMSTYLE.NACMD.WAITSECS variable
 3. The default value of 90 seconds
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NACMD command. For more information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NetID

Purpose

The NetID statement specifies the network ID as a global variable.

Syntax

The NetID statement has the following syntax:

NetID

►►—NetID—=—network_id—►►

where:

network_id

Indicates a 1- to 8-character network identifier.

Usage notes

- If you set the &CNMNETID system symbolic in the IEASYMxx member of the SYS1.PARMLIB data set, the symbol can be used in the CNMSTYLE member. This is the default.
- The common CNMSTYLE variable that is named CNMSTYLE.NETID is assigned from the NetID statement but is updated when the actual VTAM value is discovered.

NLDM.AMLUTDLY

Purpose

Use the NLDM.AMLUTDLY statement to specify the number of seconds the NetView program waits before trying again to access the domain table built by the DSIAMLUT task.

Syntax

The NLDM.AMLUTDLY statement has the following syntax:

►►—NLDM.AMLUTDLY—=—30—►►
►►—NLDM.AMLUTDLY—=—seconds—►►

where:

seconds

Specifies the number of seconds from 1–300. The default value is 30.

If you use a value that is not valid, an error message is issued and the NetView program uses the default value.

If a timeout condition occurs, the AAU085I message is issued.

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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.AUTHDOM

Purpose

Use the NLDM.AUTHDOM statement to initialize cross-domain authorization. You can code as many of these statements as needed for your environment.

Syntax

The NLDM.AUTHDOM statement has the following syntax:

```
►►NLDM.AUTHDOM.—suffix—=*ANY*→  
          | *NONE* →  
          | domainid →
```

where:

suffix

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

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

If more than one NLDM.AUTHDOM statement has the same suffix, the last statement processed will be used. NLDM will process the AUTHDOM statements in alphabetical order of the suffixes.

ANY

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

NLDM.AUTHDOM.&DOMAIN. = *ANY*

To override this initial setting in CNMSTYLE, you can code another NLDM.AUTHDOM.&DOMAIN. = *value* statement in any of the CNMSTYLE included members, such as CxxSTGEN.

ANY and *NONE* are mutually exclusive. *ANY* and *domainid* are mutually exclusive. If *ANY* is coded with *NONE* or *domainid*, the AUTHDOM statement with the higher suffix will be processed first, and will be in effect.

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. *NONE* and *ANY* are mutually exclusive. *NONE* and *domainid* are mutually exclusive. If *NONE* is coded with *ANY* or *domainid*, the AUTHDOM statement with the higher suffix will be processed first, and will be in effect.

domainid

Indicates that any operator using the specified NetView program can establish a cross-domain session (SDOMAIN) with this NetView program. The *domainid* is mutually exclusive with *ANY* and *NONE*. You can code as many NLDM.AUTHDOM.suffix=*domainid* statements as needed for your environment.

Usage notes

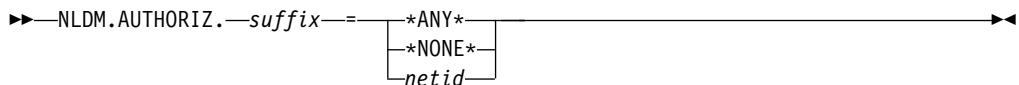
To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.AUTHORIZ**Purpose**

The NLDM.AUTHORIZ statement specifies 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.

Syntax

The NLDM.AUTHORIZ statement has the following syntax:



where:

suffix

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

NLDM.AUTHORIZ.A
NLDM.AUTHORIZ.A
NLDM.AUTHORIZ.B

If more than one NLDM.AUTHORIZ statement has the same suffix, the last statement processed will be used. NLDM will process the AUTHORIZ statements in alphabetical order of the suffixes.

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:

NLDM.AUTHORIZ.&CNMNETID. = *ANY*

To override this initial setting in CNMSTYLE, you can code another NLDM.AUTHORIZ.&CNMNETID. = *value* statement in any of the CNMSTYLE included members, such as CxxSTGEN.

ANY and *NONE* are mutually exclusive. *ANY* and *netid* are mutually exclusive. If *ANY* is coded with *NONE* or *netid*, the AUTHORIZ statement with the higher suffix will be processed first, and will be in effect.

NONE

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

NONE and *ANY* are mutually exclusive. *NONE* and *netid* are mutually exclusive. If *NONE* is coded with *ANY* or *netid*, the AUTHORIZ statement with the higher suffix will be processed first, and will be in effect.

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

netid

Specifies the network identifier. The netid is mutually exclusive with *ANY* and *NONE*. You can code up to 255 NLDM.AUTHORIZ.suffix=*netid* statements in the CNMSTYLE member as needed for your environment.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.CDRMDEF

Purpose

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), which is an 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.

Syntax

The NLDM.CDRMDEF statement has the following syntax:

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 definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

Related statements

LUC.CNMTARG, DSTINIT, NCCFID, TASK

NLDM.CDTIME

Purpose

The NLDM.CDTIME statement specifies the timeout 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 timeout value is used in the home domain in case the other domain issues a cross-domain request for data using CDTIME as the timeout value.

Syntax

The NLDM.CDTIME statement has the following syntax:

►►NLDM.CDTIME—=—*seconds*►►

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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.DRDELAY

Purpose

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.

Syntax

The NLDM.DRDELAY statement has the following syntax:

►►NLDM.DRDELAY—=—*seconds*—►►

where:

seconds

Specifies the number of seconds in the range of 1 - 60. Use values less than 8 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.DSRBO

Purpose

The NLDM.DSRBO statement specifies the number of DSRBOs to be associated with the AAUTSKLP task.

Syntax

The NLDM.DSRBO statement has the following syntax:

►►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 preallocate for processing solicited RUs and Virtual Storage Access Method (VSAM) requests. If more requests are received than DSRBs are 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.ERCOUNT

Purpose

The NLDM.ERCOUNT specifies the number of total explicit routes known to this session monitor.

Syntax

The NLDM.ERCOUNT statement has the following syntax:

►—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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.FCTIME

Purpose

The NLDM.FCTIME statement specifies 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.

Syntax

The NLDM.FCTIME statement has the following syntax:

►►NLDM.FCTIME==*seconds*►►

where:

seconds

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

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.KEEPDISC

Purpose

The NLDM.KEEPDISC statement specifies the number of discarded PIU trace data records to be kept in virtual storage.

Syntax

The NLDM.KEEPDISC statement has the following syntax:

►►NLDM.KEEPDISC==*records*►►

where:

records

Specifies the number of records in the range of 1 - 999. The default is 250.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.KEEPMEM

Purpose

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

Syntax

The NLDM.KEEPMEM statement has the following syntax:

►►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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.KEEPPIU

Purpose

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

Syntax

The NLDM.KEEPPIU statement has the following syntax:

►—NLDM.KEEPPIU—=—records—►

where:

records

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

Usage notes

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

NLDM.KEEPRTM

Purpose

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

Syntax

The NLDM.KEEPRTM statement has the following syntax:

►—NLDM.KEEPRTM—=—periods—►

where:

periods

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

Usage notes

You can change RTM keep counts only at initialization.

NLDM.KEEPSESS

Purpose

The NLDM.KEEPSESS statement specifies whether DASD session wrapping is used.

Syntax

The NLDM.KEEPSESS statement has the following syntax:

►►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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.LOG

Purpose

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.

Syntax

The NLDM.LOG statement has the following syntax:

►►NLDM.LOG=NO
NLDM.LOG=YES►►

where:

YES

Specifies to write records to the external log.

NO Specifies that records are not to 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 3 shows valid combinations for these statements.

Table 3. Combinations of NLDM.LOG and NLDM.SESSTATS

If you specify:	The NetView program writes to the external log:
• NLDM.LOG=YES • NLDM.SESSTATS=YES	<ul style="list-style-type: none">• Response time data (if SAW=YES and RTM=YES)• Configuration data• Availability and accounting data:<ul style="list-style-type: none">– Session start records, session end records, combined session start-end records– Session statistics (PIU counts)
• NLDM.LOG=YES • NLDM.SESSTATS=NO	<ul style="list-style-type: none">• Response time data (if SAW=YES and RTM=YES)• Configuration data• Combined session start-end records
NLDM.LOG=NO	No session monitor data, regardless of the SESSTATS parameter. This is the default.
• NLDM.LOG=YES • NLDM.SESSTATS=AVAIL	<ul style="list-style-type: none">• Response time data (if SAW=YES and RTM=YES).• Configuration data• Availability data (if a KCLASS statement specifies, or defaults to, AVAIL=YES) Availability data includes session start records, session end records, and combined session start-end records.

For more information about the record formats that the NetView program writes to the external log, refer to *IBM Tivoli NetView for z/OS Application Programmer's Guide*.

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.LUCOUNT

Purpose

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.

Syntax

The NLDM.LUCOUNT statement has the following syntax:

►►NLDM.LUCOUNT—=—*number*————►►

where:

number

Specifies a value in the range of 1 - 999999.

When you assign a value, include all LUs owned by the local system services control point (SSCP) and all LUs owned by other SSCPs that can have session partners owned by the local SSCP.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.MACRF

Purpose

The NLDM.MACRF statement specifies the local shared resource (LSR) options.

Syntax

The NLDM.MACRF statement has the following syntax:

►►NLDM.MACRF—=—

LSR
DFR
NSR

————►►

where:

LSR

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

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.

Note: Do not use the DFR option unless instructed by IBM Software Support.

NSR

Indicates that the data set does not use shared resources.

Note: Do not use the NSR option unless instructed by IBM Software Support.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.MAXEND

Purpose

The NLDM.MAXEND statement specifies the number of concurrent requests for PIU trace data.

Syntax

The NLDM.MAXEND statement has the following syntax:

►►—NLDM.MAXEND—=—*number*—►►

where:

number

Specifies a value in the range of 1 - 999.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.NETID

Purpose

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

Syntax

The NLDM.NETID statement has the following syntax:

►►—NLDM.NETID—=—*netid*—►►

where:

netid

Specifies the 1 - 8 character name of this network.

NETID must be defined consistently in your network. Each NetView program or session monitor in a network must have the same NETID, whether it is defined to the NetView program, to the session monitor, or to VTAM.

Usage notes

- The NLDM.NETID statement is commented out in the CNMSTYLE member.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.PDDNM

Purpose

The NLDM.PDDNM statement specifies the session monitor primary data set.

Syntax

The NLDM.PDDNM statement has the following syntax:

►►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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.PERFMEM

Purpose

The NLDM.PERFMEM statement specifies the DSIPARM member name for performance class definitions.

Syntax

The NLDM.PERFMEM statement has the following syntax:

►►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 the CNMSTYLE member.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.PEXLST xx

Purpose

The NLDM.PEXLST xx statement specifies an exception list to be used with the following commands:

- DBAUTO
- DBINIT
- DBMAINT
- NLDM PURGE
- PURGEDB

You can code as many of these statements as needed for your environment.

Syntax

The NLDM.PEXLST xx statement has the following syntax:

```
➡➡NLDM.PEXLST $xx$ .suffix=—resname1—resname2  
                          └session_type┘
```

where:

xx Specifies a unique value consisting of two alphanumeric or national (@, #, or \$) characters. In the following example, 01 and 02 are used:

NLDM.PEXLST**01**.B
NLDM.PEXLST**02**.A

suffix

Specifies a unique value consisting of 1 - 8 alphanumeric or national (@, #, or \$) characters. In the following example, A and B are used for the suffix:

NLDM.PEXLST**01**.A
NLDM.PEXLST**01**.B

resname1 resname2

Specifies the resource names of the session partner data. These names must be 1 - 8 alphanumeric characters and can contain the wildcard characters * and ?.

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 exist in the position of a ?; that is, A?B is not matched by AB because no character replaces 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.

session_type

Specifies session data. The following sessions are valid:

- CP-CP

- LU-LU
- SSCP-LU
- SSCP-PU
- SSCP-SSCP

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

Examples

- Exception list PEXLST01 excludes any NCP* resources or any SSCP-SSCP sessions:

```
NLDM.PEXLST01.A = NCP* *
NLDM.PEXLST01.B = SSCP-SSCP
```

- Exception list PEXLST02 excludes any sessions between HOST1 and NCP* resources, or any CP-CP sessions:

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

NLDM.PIUTNUM

Purpose

The NLDM.PIUTNUM statement specifies the number of PIU trace data buffers.

Syntax

The NLDM.PIUTNUM statement has the following syntax:

►—NLDM.PIUTNUM—=—*number*—►

where:

number

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

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.PIUTSIZE

Purpose

The NLDM.PIUTSIZE specifies the size of the PIU trace data buffers.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.RETRY

Purpose

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.

Syntax

The NLDM.RETRY statement has the following syntax:



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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

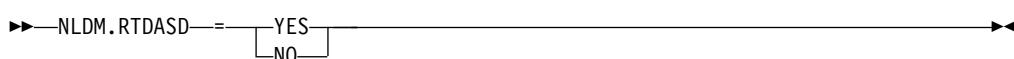
NLDM.RTDASD

Purpose

The NLDM.RTDASD statement indicates whether to write ER data to the database.

Syntax

The NLDM.RTDASD statement has the following syntax:



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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.RTM

The NLDM.RTM statement enables the NetView response time monitor (RTM) function.

The NLDM.RTM statement has the following syntax:



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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.RTMDISP

Purpose

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

Syntax

The NLDM.RTMDISP statement has the following syntax:



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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.SAW

Purpose

The NLDM.SAW statement specifies whether session awareness (SAW) data is collected.

Syntax

The NLDM.SAW statement has the following syntax:



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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command.

NLDM.SAWNUM

Purpose

The NLDM.SAWNUM statement specifies the number of session awareness (SAW) data buffers.

Syntax

The NLDM.SAWNUM statement has the following syntax:



where:

number

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

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.SAWSIZE

Purpose

The NLDM.SAWSIZE statement specifies the size of session awareness (SAW) data buffers.

Syntax

The NLDM.SAWSIZE statement has the following syntax:

►►NLDM.SAWSIZE==*buffer_size*►►

where:

buffer_size

Specifies the buffer size in a range 2K - 32K. The initial setting in the CNMSTYLE member is 4K.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.SDDNM

Purpose

The NLDM.SDDNM statement specifies session monitor secondary data set.

Syntax

The NLDM.SDDNM statement has the following syntax:

►►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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.SESSIONSMAX

Purpose

The NLDM.SESSIONSMAX statement specifies the maximum number of sessions that can be displayed by the SESS command.

Syntax

The NLDM.SESSIONSMAX statement has the following syntax:

►—NLDM.SESSIONSMAX—=—*number*—►

where:

number

Specifies a value in the range of 1 - 999999. The initial setting in the CNMSTYLE member is 999.

Usage notes

- By specifying a high *number*, you might increase the amount of storage and VSAM resources used by the SESS command.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.SESSTATS

Purpose

The NLDM.SESSTATS statement specifies whether the network accounting and availability functions are active.

Syntax

The NLDM.SESSTATS statement has the following syntax:

►—NLDM.SESSTATS=*NO*—►

►—NLDM.SESSTATS=*AVAIL*—►

 |

 |—NLDM.SESSTATS=*NO*—►

 |

 |—NLDM.SESSTATS=*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 3 on page 155 shows valid combinations for these statements.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.TRACEGW

Purpose

The NLDM.TRACEGW statement specifies whether the NetView program activates gateway tracing at NetView program initialization or NCP activation.

Syntax

The NLDM.TRACEGW statement has the following syntax:



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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

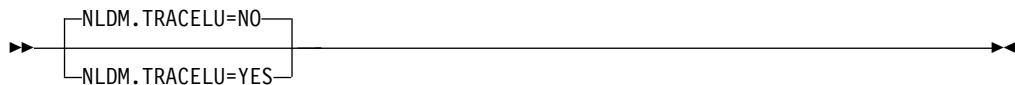
NLDM.TRACELU

Purpose

The NLDM.TRACELU statement specifies whether the NetView program starts tracing LU information at NetView program initialization.

Syntax

The NLDM.TRACELU statement has the following syntax:



where:

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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NLDM.TRACESC

Purpose

The NLDM.TRACESC statement specifies whether the NetView program starts tracing SSCP information at NetView program initialization.

Syntax

The NLDM.TRACESC statement has the following syntax:



where:

NO Indicates that the NetView program does not start tracing SSCP information until the TRACE START command is entered. This is the default value.

YES

Indicates that the NetView program starts tracing SSCP information at NetView program initialization.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NLDM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NMCstatus.errorDSN

Purpose

The NMCstatus.errorDSN statement defines a data set that contains any errors found in the GRAPHICS policy file.

Syntax

The NMCstatus.errorDSN statement has the following syntax:

►►—NMCstatus.errorDSN—=—*data_set_name*————►►

where:

data_set_name

Data set name. You can specify *NONE* if no data set is needed.

Usage notes

- Many of the messages are greater than 80 characters in length so if a FIXED 80 file is specified, most messages are truncated.
- For more information, see “NMCSTATUS (Control File Entry)” on page 353.

NPDA.ALCACHE

Purpose

The NPDA.ALCACHE statement specifies the maximum size of your alert cache.

Syntax

The NPDA.ALCACHE statement has the following syntax:

NPDA.ALCACHE

►►—NPDA.ALCACHE = 10————►►

►►—NPDA.ALCACHE = —*maxnumber*————►►

 |———*maxnumber*————►►

 |———NONE————►►

 |———WRAPCNT————►►

where:

10|*maxnumber*

Specifies the maximum number of alert records that are kept in storage. The 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 the Alerts-Dynamic panel, the alert can be retrieved from the alert cache.

Usage notes

- 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.

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).

 - 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.
- Refer to *IBM Tivoli NetView for z/OS Automation Guide* for more information.
- The alert cache is set to the smaller of one of the following values:
 - 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.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Examples

Table 4. Examples of the NPDA.ALCACHE Statement

ALCACHE Statement	Alert Wrap Count	Alert Cache Count	Comments
NPDA.ALCACHE = 500	100	100	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.
NPDA.ALCACHE = 500	500	500	Because the alert wrap count is the same size as the value specified with ALCACHE, the alert cache size is 500.
NPDA.ALCACHE = 100	999	100	Because the value specified with ALCACHE is smaller than the alert wrap count, the alert cache size is 100.

Table 4. Examples of the NPDA.ALCACHE Statement (continued)

ALCACHE Statement	Alert Wrap Count	Alert Cache Count	Comments
NPDA.ALCACHE = NONE	100	0	The alert cache is not used to store alerts.
NPDA.ALCACHE = WRAPCNT	500	500	The alert cache is set to the size of the alert wrap count.

Related statements

NPDA.ALERTFWD

NPDA.ALERTFWD

Purpose

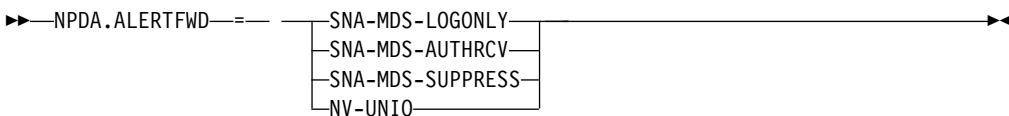
The NPDA.ALERTFWD statement defines the alert forwarding protocol that this NetView host is to use.

For more information about alert forwarding, refer to the *IBM Tivoli NetView for z/OS Automation Guide*.

Syntax

The NPDA.ALERTFWD statement has the following syntax:

NPDA.ALERTFWD



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*). Use SNA-MDS except when the focal point is before 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 DS16DST task initialization member DS16INIT 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 DEFFOCPCT statements for category ALERT in the DSICRTR task initialization member DSICRTTD are accepted and processed. Any DEFFOCPCT 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 *IBM Tivoli NetView for z/OS Automation Guide* 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

DEFFOCPT, NPDA.ALCACHE, NPDA.ALRTINFP

NPDA.ALERTLOG

Purpose

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 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 the CNMSTYLE member.

Syntax

The NPDA.ALERTLOG statement has the following syntax:

NPDA.ALERTLOG

►—NPDA.—ALERTLOG— —=— RANDRANG
RSTD RANG

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,S).

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 RSTDRA NG 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 RSTDRA NG setting if you find that hardware monitor initialization is taking too long to complete with the default RANDRANG setting.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.ALRTINFP

Purpose

The NPDA.ALRTINFP statement specifies whether SNA-MDS/LU 6.2 forwarded alert records and whether event/statistic records for non-NetView entry points are 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 used to forward alerts. The NPDA.ALERTFWD statement with the SNA-MDS/LU 6.2 option must be defined in the CNMSTYLE member.
2. Alerts must be received from another node (either a NetView or non NetView node) 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 the CNMSTYLE member. For more information on architected alert forwarding, refer to the *IBM Tivoli NetView for z/OS Automation Guide*.

Syntax

The NPDA.ALRTINFP statement has the following syntax:

NPDA.ALRTINFP

```
►►—NPDA—.—ALRTINFP—.—RECORD— —=— YES  
                                                  NO
```

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. This is the initial setting in the CNMSTYLE member.

- 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 hardware monitor database of the intermediate node 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 the alert focal point of an intermediate node 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 the CNMSTYLE member.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

NPDA.ALERTFWD, DEFFOCPT

NPDA.ALT_ALERT

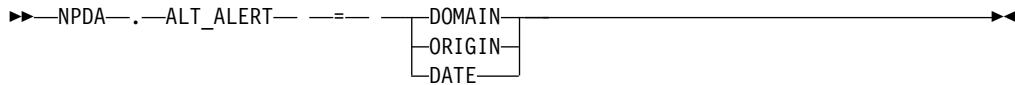
Purpose

The NPDA.ALT_ALERT statement defines the data that is displayed on the Alerts-Dynamic (NPDA-30A), Alerts-Static (NPDA-30B), and Alerts-History (NPDA-31A) hardware monitor panels.

Syntax

The NPDA.ALT_ALERT statement has the following syntax:

NPDA.ALT_ALERT



where:

DOMAIN

Displays the name of the domain that originally received the alert. This is the default value in the CNMSTYLE member.

ORIGIN

Displays the name of the resource sending the alert. If the alert has been forwarded from an entry point, you see the domain name of the entry point NetView program, preceded by an asterisk (*).

DATE

Displays the date and time the alert was recorded in the alert database.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.AUTORATE

Purpose

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 182 for more information.

Syntax

The NPDA.AUTORATE statement has the following syntax:

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 the CNMSTYLE member.
- 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.

Usage notes

- If you do not code an NPDA.AUTORATE statement, MSUs blocked by a filter set by the RATE function are not automated.
- XITCI return code 252 indicates that an MSU is only logged to the system management facilities (SMF). 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.
- Make updates to the NPDA.AUTORATE statement in the CNMSTUSR or CxxSTGEN member. Changes do not take effect until you run the RESTYLE command with the NPDA option. If you change the NPDA.AUTORATE statement and the change results in additional extents being used, stop and restart the NetView program to implement the change. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

NPDA.RATE

NPDA.DSRBO

Purpose

The NPDA.DSRBO statement specifies the number of DSRBOs to be associated with the BNJDSERV task.

Syntax

The NPDA.DSRBO statement has the following syntax:

►►—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 preallocate for processing solicited RUs and VSAM requests. If more requests are received than DSRBs are 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.DSRBU

Purpose

The NPDA.DSRBU statement specifies the number of data services request blocks (DSRBs) to pre-allocate for processing unsolicited problem determination request units (RUs).

Syntax

The NPDA.DSRBU statement has the following syntax:

►►—NPDA.DSRBU—=—number—►►

where:

number

Is a decimal number from 0 - 999. The default value is 5.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.ERR_RATE

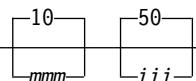
Purpose

Use the NPDA.ERR_RATE statement to regulate the generation of messages for alerts that are not valid.

Syntax

The NPDA.ERR_RATE statement has the following syntax:

NPDA.ERR_RATE

►►—NPDA—.—ERR_RATE— —=— —►►

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|i ii

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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.MACRF

Purpose

The NPDA.MACRF statement specifies the local shared resource (LSR) options.

Syntax

The NPDA.MACRF statement has the following syntax:



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 notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.MDSIND

Purpose

The NPDA.MDSIND statement specifies whether the at sign (@) is displayed on alert panels to indicate that an alert was forwarded by the SNA-MDS method.

Syntax

The NPDA.MDSIND statement has the following syntax:

```
►—NPDA.MDSIND—= YES  
                  | NO
```

where:

YES

Specifies that the at sign is displayed.

NO Specifies that the at sign is not displayed.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.PDDNM

Purpose

The NPDA.PDDNM statement specifies the hardware monitor primary data set.

Syntax

The NPDA.PDDNM statement has the following syntax:

```
►—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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.PDFILTER

Purpose

NPDA.PDFILTER statements specify commands that are run when the hardware monitor BNJDSERV task starts, for example recording filters (SRFILTER commands).

Syntax

The NPDA.PDFILTER statement has the following syntax:

NPDA.PDFILTER

►—NPDA—.—PDFILTER—.—*suffix*— —=— —*command*————→

where:

suffix

A required identifier that makes an NPDA.PDFILTER statement unique. Use a different suffix for each statement coded.

command

The command to be run when the hardware monitor BNJDSERV task starts.

Usage notes

- The commands specified by NPDA.PDFILTER statements are called by the PDFILTER (CNME3004) command list from the automation table DSITBL01 when the BNJDSERV task starts.
- Do not add hardware monitor viewing filters (SVFILTER commands) because they are only effective for the operator who issues them.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.PRELOAD_BER

Purpose

The NPDA.PRELOAD_BER data is used if your network or system contains devices or applications that send Basic Encoding Rules (BER) data to the NetView program.

Syntax

The NPDA.PRELOAD_BER statement has the following syntax:

NPDA.PRELOAD_BER

►—NPDA.PRELOAD_BER = NO————→
 |————NPDA.PRELOAD_BER = YES————→

where:

YES

Specifies that PRELOAD_BER data is sent to the NetView program.

NO Specifies that PRELOAD_BER data is not sent to the NetView program. This is the default.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.R (Ratio)

Purpose

The NPDA.R (Ratio) statement assigns 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 conditions 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.

Syntax

The R statement has the following syntax:

NPDA.R (RATIO)

►—NPDA—.—R—.—suffix— —=— —rtype— —rname— —vvv————►

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 in the following ways:

- 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 processor

LDEV Is a device attached to the remote cluster controller

LINK_ATT

Is a link-attached resource

CHAN_ATT

Is a channel-attached resource

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 (*).

vvv

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.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. If changing CNMYSTYLE results in additional extents being used, stop and restart the NetView program. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

NPDA.W

NPDA.RATE

Purpose

The NPDA.RATE statement sets 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 because of a damaged database (caused by setting the wrong date when initializing the system).

Syntax

The NPDA.RATE statement has the following syntax:

NPDA.RATE

►—NPDA—.—RATE— —— mm:ss—————►

where:

mm:ss

Identifies the number of minutes and seconds that must elapse before an event 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 the CNMSTYLE member. See “NPDA.AUTORATE” on page 175 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

NPDA.AUTORATE, NPDA.W

NPDA.REPORTS

Purpose

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.

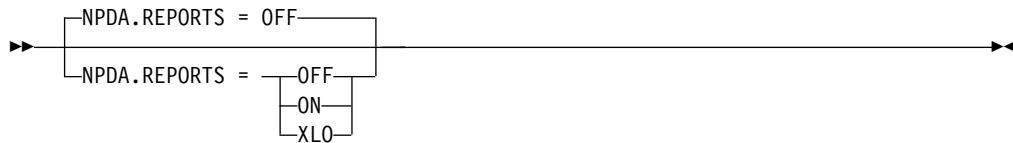
Change the report-logging setting any time after initialization by issuing the REPORTS command.

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 *IBM Tivoli NetView for z/OS Application Programmer’s Guide*.

Syntax

The NPDA.REPORTS statement has the following syntax:

NPDA.REPORTS



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 conditions 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.

XL0

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 conditions 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

DSTINIT, TASK

NPDA.SDDNM

Purpose

The NPDA.SDDNM statement specifies hardware monitor secondary data set.

Syntax

The NPDA.SDDNM statement has the following syntax:

►►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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.TECROUTE

Purpose

The NPDA.TECROUTE statement specifies the name of the PPI receiver associated with the Event/Automation Service (E/AS).

Syntax

The NPDA.TECROUTE statement has the following syntax:

►►—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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NPDA.W (Wrap)

Purpose

The NPDA.W (Wrap) statement assigns initial wrap values for alerts on the hardware monitor database. 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 (*nn*) 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 member. These values are called from CNMSTYLE member by the BNJMBDST sample when the NPDA.W statement is run.

Event and statistical wrap count statements must indicate the resource type (*rtype*). The alert (AL) statement in the CNMSTYLE member sets wrap counts for alerts on the hardware monitor and NPDA database.

The NPDA.W statements assign initial wrap values when the first error record is added to the Alert History Record on the database. After the initial wrap values are set, the NPDA.W statement is not read again.

After this statement is read and alerts are on the database, the NPDA.W statement is not read again. Cycling the NetView program or the hardware monitor after this point does not read the NPDA.W statements in the CNMSTYLE member.

To change the wrap values for existing resources, use the SWRAP command (or reinitialize the hardware monitor database). Refer to the NetView online help for more information about the SWRAP command. The altered values remain in effect until one of the following conditions occurs:

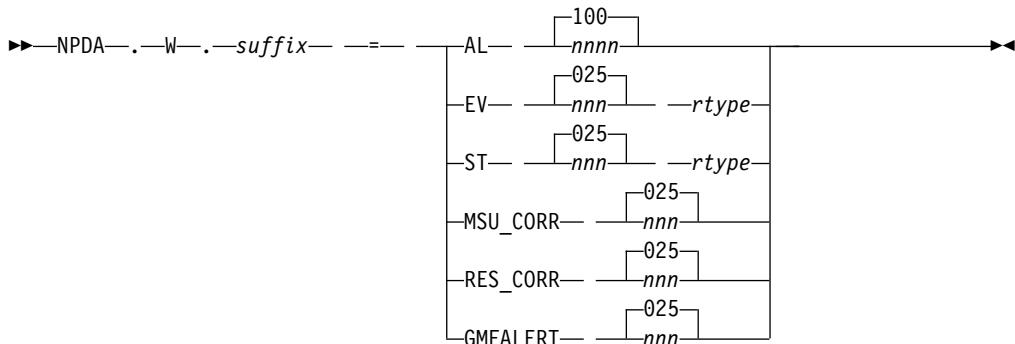
- The hardware monitor database is reinitialized.
- All hardware monitor records for the particular resource and all attached resources are purged.
- The values are changed by another SWRAP command.

The MSU, resource correlation, and GMFALERT wrap counts cannot be changed by the SWRAP command.

Syntax

The NPDA.W statement has the following syntax:

NPDA.W (Wrap)



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.

025|nnn

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.

025|nnn

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.

025|nnn

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.
- Specifying a wrap count for *rtype* only affects events for which the lowest (rightmost) level of the alert hierarchy matches the value of *rtype*.
- An NPDA.W statement takes effect only if no associated records on the hardware monitor database exist. If associated records exist, NPDA.W settings are ignored.

For example, if the hardware monitor database contains alert records and the AL wrap count in effect when the last alert was recorded was 100, you cannot use the NPDA.W statement to reset the wrap count. To change the wrap count, use the NPDA SWRAP command.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NPDA command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Related statements

TASK, NPDA.RATE

NQNSUP.USE

Purpose

The NQNSUP.USE statement controls whether the NetView Management Services transport running on this NetView program is to use network qualified names on all requests to send and receive Management Services data.

Syntax

The NQNSUP.USE statement has the following syntax:

NQNSUP.USE



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 notes

If you change the value of NQNSUP.USE after the NetView program is initialized, the NetView program must be recycled for changes to take effect.

NRM.CMODE

Purpose

The NRM.CMODE statement specifies the mode to communicate between NetView hosts for the NetView Resource Manager function.

Syntax

The NRM.CMODE statement has the following syntax:

NRM.CMODE



where:

number

A non-zero numerical value, indicating that multiple NRM.CMODE values can be specified. The first NRM.CMODE statement must start with the number 1, and any additional statements must be in ascending order (2, 3, 4, ...) with no gaps in numbering.

IP Indicates to use TCP over IPv4 communication mode.

IPV6

Indicates to use TCP over IPv6 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 specify one value of SNA, IP, or IPV6. If multiple communication modes are used, specify an NRM.CMODE statement for each NRM.HOSTDEST statement.

- NRM.CMODE is ignored if NRM.HOSTDEST is not specified.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- To communicate with a version of NetView prior to V5R3 (NetView Resource Manager agent or manager), specify SNA or IP for the communication mode. IPV6 is not supported.

NRM.CONNINT

Purpose

The NRM.CONNINT statement specifies the interval between connection retries when establishing communication between the manager and agent hosts.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.CONNINT statement has the following syntax:

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 with NRM.CONNTRY.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.CONNTRY

Purpose

The NRM.CONNTRY statement specifies the number of connection retries to be attempted when communication between the manager and agent fails.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.CONNTRY statement has the following syntax:

NRM.CONNRETRY

►►—NRM.CONNRETRY—=—*retries*—►►

where:

retries

Specifies a value is in the range 0 - 10. The default is 3.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.HBRETRY

Purpose

The NRM.HBRETRY statement specifies how often a manager tests connectivity to its agents.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.HBRETRY statement has the following syntax:

NRM.HBRETRY

►►—NRM.HBRETRY—=—*seconds*—►►

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 the NetView Resource Manager 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.HOSTDEST

Purpose

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 used for the NetView Resource Manager function.

Syntax

The NRM.HOSTDEST statement has the following syntax:

NRM.HOSTDEST

►►—NRM.HOSTDEST.—*number*—=—*destination*————►►

where:

number

A non-zero numerical value, indicating that multiple NRM.HOSTDEST values can be specified. The first NRM.HOSTDEST statement must start with the number 1, and any additional statements must be in ascending order (2, 3, 4, and so on) with no gaps in numbering.

destination

If the communication mode between this agent and the manager host is SNA (see “NRM.CMODE” on page 189), the value for *destination* is the NetView domain of the manager. If the communication mode between this agent and the manager host is IP or IPV6, the value for *destination* is an 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.IPV6SRC

Purpose

The NRM.IPV6SRC statement provides local host information that the NetView Resource Manager agent reports to a host destination (HOSTDEST) when CMODE=IPV6 is specified.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.IPV6SRC statement has the following syntax:

NRM.IPV6SRC

►►—NRM.IPV6SRC—=—*ipv6_source*————►►

where:

ipv6_source

Specifies a host name or IPv6 address. If a host name is specified, the NetView Resource Manager requests its resolution to an IPv6 address.

Usage notes

- The NRM.IPV6SRC statement is ignored if there is not a host destination that uses a communication mode of IPV6 (see “NRM.CMODE” on page 189) or if the IPv6Env statement is set to NONE.
- If you do not specify an NRM.IPV6SRC statement and there is a host destination for which CMODE=IPV6 is specified, the NetView Resource Manager function attempts to obtain the host name defined for the TCP/IP stack associated with the NetView Resource Manager. If the host name exists, the NetView Resource Manager function asks the resolver to return an IPv6 address. The resultant IPv6 address is sent to the applicable NetView Resource Manager function manager or managers.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.PORT

Purpose

The NRM.PORT statement specifies the port number for the DSIRTTR task at each NRM.HOSTDEST.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.PORT statement has the following syntax:

NRM.PORT

►►—NRM.PORT.—*number*—=—*port_number*—————►►

where:

number

A non-zero numerical value, indicating that multiple NRM.PORT values can be specified. The first NRM.PORT statement must start with the number 1, and any additional statements must be in ascending order (2, 3, 4, and so on) with no gaps in numbering.

port_number

Specify a value in the range 1 - 65535. The default is 4021.

Usage notes

- Specify only one PORT value per line.
- If only one port number is specified, that port number is used for all IP or IPV6 host destinations.
- The NRM.PORT statement is only valid when the communication mode (see “NRM.CMODE” on page 189) is set to IP or IPV6. If you specify NRM.HOSTDEST with a communication mode of SNA, leave the NRM.PORT statement commented out.

- If all the NRM.HOSTDEST statements use a communication mode of IP or IPV6 and you want to use the same port value for all the NRM.HOSTDEST statements, you can use the following NRM.PORT statement:
`NRM.PORT.1=value`
- If multiple communication modes are used, specify an NRM.PORT value for each NRM.HOSTDEST value that uses IP or IPV6 for the communication mode.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.RESET

Purpose

The NRM.RESET statement specifies if you want all NetView Resource Manager objects set to an initial status at NetView Resource Manager initialization.

Syntax

The NRM.RESET statement has the following syntax:

NRM.RESET



where:

NO Indicates that no initial status is to be set during NetView Resource Manager initialization. This is the default.

Note: NetView Resource Manager objects are to be updated to their correct status when an agent communicates status to the manager.

YES

Indicates that the status of objects is set to the status specified by NRM.STATUS.RESET under the DISPLAY STATUS section in the CNMSTYLE member.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.SAMPLERATE

Purpose

The NRM.SAMPLERATE statement specifies the frequency in requesting complete data for this NetView Resource Manager host.

Syntax

The NRM.SAMPLERATE statement has the following syntax:

NRM.SAMPLERATE

►—NRM.SAMPLERATE—=—*minutes*—►

where:

minutes

Specifies the number of minutes and is in the range 0 - 1440. The default is 0, no sampling.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.STATUS

Purpose

The NRM.STATUS statement specifies the display status values that are used for NetView Resource Manager objects.

Syntax

The NRM.STATUS statement has the following syntax:

NRM.STATUS

►—NRM.STATUS.—*state*—=—*status*—►

where:

state

Indicates the NetView Resource Manager object state. For more information, refer to the CNMSTYLE member.

status

Indicates the display status value. For more information, refer to the CNMSTYLE member.

Usage notes

To implement definition changes while the NetView program is active, change the statement in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.TYPE

Purpose

The NRM.TYPE statement specifies whether this host is a manager or an agent host.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.TYPE statement has the following syntax:

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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.XCLDOM

Purpose

The NRM.XCLDOM statement specifies which domains the manager host is not to monitor.

This statement is used for the NetView Resource Manager function.

Syntax

The NRM.XCLDOM statement has the following syntax:

NRM.XCLDOM



where:

domainid

Specifies a domain name with a value 1 - 5 characters in length. Wildcard characters 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 the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.XCLSRC

Purpose

The NRM.XCLSRC statement specifies which set of exclusion lists, if any, that you want to use with the NetView Resource Manager.

Syntax

The NRM.XCLSRC statement has the following syntax:

NRM.XCLSRC



where:

CTL

Indicates to use the exclusion lists that were already in effect. Use this to retain exclusion lists that you have defined dynamically across NetView Resource Manager calls.

VARS

Indicates to use the exclusion lists as defined in the CNMSTYLE member. Specifying VARS overrides any exclusion lists that were previously set. This is the default.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.XCLTASKN

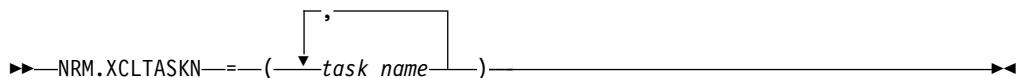
Purpose

The NRM.XCLTASKN statement specifies which tasks are not to be monitored by the NetView Resource Manager.

Syntax

The NRM.XCLTASKN statement has the following syntax:

NRM.XCLTASKN



```
►►—NRM.XCLTASKN—=—(—task_name—)————►►
```

A syntax diagram for the NRM.XCLTASKN command. It starts with a double right-pointing arrow (►►). The command name 'NRM.XCLTASKN' is followed by a single equals sign (=). Then there is a left parenthesis ((). Inside the parentheses, the word 'task_name' is underlined. Above the 'task_name' underlined text, there is a bracket with a downward arrow pointing to it, and a comma (,) is positioned above the bracket. After the 'task_name' underlined text, there is another right parenthesis) . Finally, there is a long horizontal line ending in a double right-pointing arrow (————►►).

where:

task_name

Specifies a task name 1 - 8 characters in length. Wildcard characters 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 NetView Resource Manager agent and is only applicable on the local NetView host.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NRM.XCLTASKT

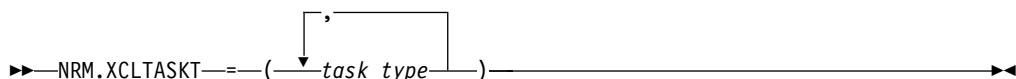
Purpose

The NRM.XCLTASKT statement specifies which types of tasks the NetView Resource Manager is not to monitor. For example, specifying NRM.XCLTASKT=OST excludes all operator station tasks from monitoring by NetView Resource Manager.

Syntax

The NRM.XCLTASKT statement has the following syntax:

NRM.XCLTASKT



```
►►—NRM.XCLTASKT—=—(—task_type—)————►►
```

A syntax diagram for the NRM.XCLTASKT command. It starts with a double right-pointing arrow (►►). The command name 'NRM.XCLTASKT' is followed by a single equals sign (=). Then there is a left parenthesis ((). Inside the parentheses, the word 'task_type' is underlined. Above the 'task_type' underlined text, there is a bracket with a downward arrow pointing to it, and a comma (,) is positioned above the bracket. After the 'task_type' underlined text, there is another right parenthesis) . Finally, there is a long horizontal line ending in a double right-pointing arrow (————►►).

where:

task_type

The following task types are valid:

- DST
- HCT
- MNT
- NNT
- OPT
- OST
- PPT

Usage notes

- The parentheses are not required if only one task type is specified. Multiple values must be enclosed in parentheses and separated by either blanks or commas.
- NRM.XCLTASKT is a function of the NetView Resource Manager agent and is only applicable on the local NetView host.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NRM command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVROLE

Purpose

The NVROLE statement describes a particular function being performed by the NetView program. This data is displayed in user interfaces.

Syntax

The NVROLE statement has the following syntax:



where:

GDPS

Geographically Dispersed Parallel Sysplex® (GDPS®).

NETWORK

Network.

SA System Automation for z/OS.

user_data

A 16-character text string with the NetView function description.

Usage notes

- The value of the NVROLE statement is used for display purposes only.
- The NVROLE statement sets the CNMSTYLE.NVROLE common global variable.

NVSP.*servername*.NUMTHRDS

Purpose

The NVSP.*servername*.NUMTHRDS statement specifies the number of threads that are created to service incoming and outgoing IP connections. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.*servername*.NUMTHRDS statement has the following syntax:

NVSP.*servername*.NUMTHRDS



where:

servername

Specifies the NetView Web Services server name.

nnn

Specifies the number of threads. The minimum value is 1. Values greater than 2147483647 default to 2147483647.

Usage notes

- The value for NUMTHRDS is assigned in the following order:
 1. The NUMTHRDS value specified with the SOACTL command.
 2. The NVSP.*servername*.NUMTHRDS variable in the CNMSTYLE member.
 3. The default value is 20.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVSP.*servername*.PDS

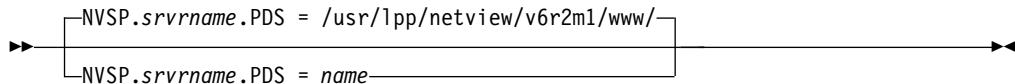
Purpose

The NVSP.*servername*.PDS statement specifies the location where server web resources files are located. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.*servername*.PDS statement has the following syntax:

NVSP.srvrname.PDS



where:

srvrname

Specifies the NetView Web Services server name.

name

Specifies the PDS name or USS directory where server initialization files are located.

Usage notes

- The value for PDS is assigned in the following order:
 1. The PDS or directory value specified with the SOACTL command.
 2. The NVSP.srvrname.PDS variable in the CNMSTYLE member.
 3. The default value is /usr/lpp/netview/v6r2m1/www/.
- If you change the PDS default subdirectory, copy all the files in the existing subdirectory to the new location, keeping the same file permissions.
- Specify the MVS-based PDS as //*base-pds*. To use GIF or JPG images in an HTML file, store the image as *base-pds.GIF(imagename)* or *base-pds.JPG(imagename)*.

An example follows:

- PDS specification on the SOACTL command or on the NVSP.srvrname.PDS statement: //'USER.INIT'
- HTML file location: USER.INIT(MYHTML)
- GIF image location: USER.INIT.GIF(MYGIF)
- HTML source (image URL address reference): GIF(MYGIF)
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVSP.srvrname.PORT

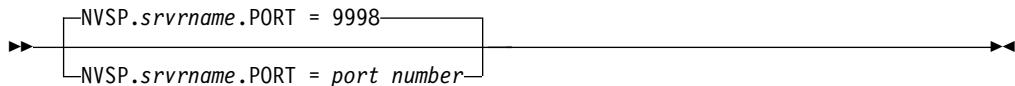
Purpose

The NVSP.srvrname.PORT statement specifies the port where the Web Services server is listening for connection requests. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.srvrname.PORT statement has the following syntax:

NVSP.srvrname.PORT



where:

srvrname

Specifies the NetView Web Services server name.

port_number

Specifies the port number. The minimum value is 0. Values greater than 2147483647 default to 2147483647.

Usage notes

- The value for PORT is assigned in the following order:
 1. The PORT value specified with the SOACTL command.
 2. The NVSP.srvrname.PORT variable in the CNMSTYLE member.
 3. The default value is 9998.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVSP.srvrname.SECURE

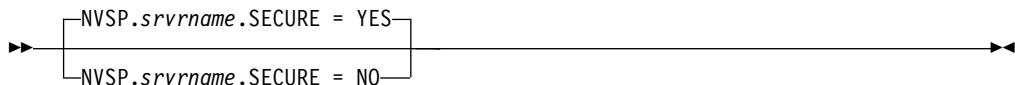
Purpose

The NVSP.srvrname.SECURE statement specifies whether the transport is secured with Application-Transparent Transport Layer Security (AT-TLS) encryption. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.srvrname.SECURE statement has the following syntax:

NVSP.srvrname.SECURE



where:

srvrname

Specifies the NetView Web Services server name.

NO Specifies that the transport is not secured.

YES

Specifies that the transport is secured.

Usage notes

- If you use the Web Services Gateway function with secure communications enabled, you must configure AT-TLS to secure inbound connections to the server. If AT-TLS is not configured for the NetView Web Services server and the CNMSTYLE NVSP.srvrname.SECURE statement is set to YES, all inbound connections are rejected. See information about controlling the access to the Web Services Gateway function in the *Security Reference*.
- The value for SECURE is assigned in the following order:
 1. The SECURE value specified with the SOACTL command.
 2. The NVSP.srvrname.SECURE variable in the CNMSTYLE member.
 3. The default value is YES.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVSP.srvrname.TRC

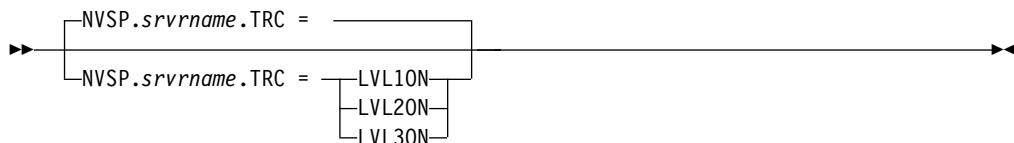
Purpose

The NVSP.srvrname.TRC statement specifies the trace levels. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.srvrname.TRC statement has the following syntax:

NVSP.srvrname.TRC



where:

srvrname

Specifies the NetView Web Services server name.

LVL10N

Starts level 1 trace.

LVL20N

Starts level 1 and 2 traces.

LVL30N

Starts level 1, 2, and 3 traces.

Usage notes

- The value for TRC is assigned in the following order:
 1. The TRC value specified with the SOACTL command.
 2. The NVSP.srvrname.TRC variable in the CNMSTYLE member.
 3. The default value is blank, which is all trace levels turned off.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

NVSP.*srvrname*.WAIT

Purpose

The NVSP.*srvrname*.WAIT statement specifies the time to wait for a command response. This statement is used with the Web Services Gateway function.

Syntax

The NVSP.*srvrname*.WAIT statement has the following syntax:

NVSP.*srvrname*.WAIT



where:

srvrname

Specifies the NetView Web Services server name.

seconds

Specifies the time interval in the range of 1 - 10000000 seconds. Values greater than 10000000 default to 10000000. The default is 30 seconds.

Usage notes

- If you specify 0 seconds (no wait), the default value of 30 seconds is used.
- The value for WAIT is assigned in the following order:
 1. The WAIT value specified with the SOACTL command.
 2. The NVSP.*srvrname*.WAIT variable in the CNMSTYLE member.
 3. The default value is 30.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE NVSP command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

OpDsPrefix

Purpose

The OpDsPrefix statement sets the common global variable OpDsPrefix to your operator data set prefix.

Syntax

The OpDsPrefix statement has the following syntax:

OpDsPrefix

►►—OpDsPrefix—=—prefix—►►

where:

prefix

Specifies the data set prefix. The default naming convention is NETVIEW.OPDS

Usage notes

- For more information about defining operator data sets, refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE OPDSPREFIX command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

OPKT.STORAGE

Purpose

The OPKT.STORAGE statement specifies the number of megabytes of data space storage allocated to OSA packet trace data collected by the NetView program.

Syntax

The OPKT.STORAGE statement has the following syntax:

OPKT.STORAGE

►►—OPKT.STORAGE.*stackname*=*storg*—►►

where:

stackname

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

storg

The number of megabytes of data space storage to allocate. This value must be a positive or negative integer, followed by the letter M. A positive value indicates that the OSA packet storage wraps. That is, when the storage area is full, arriving packets cause the oldest packets to be purged. A negative value indicates that the OSA packet storage does not wrap. That is, when the storage area is full, the collection process stops, as in STOPCOLL (see the online help for the PKTS command for more information). The minimum integer value is 16M, and the maximum integer value is 2047M. The default value is 16M.

Usage notes

This statement applies only if collection of packet trace data is started automatically during NetView initialization using the INIT.OPKT statement. To specify storage allocation when starting data collection manually, use the STORAGE key word on the PKTS START command. (See the *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)* for more information.)

PKTS.STORAGE

Purpose

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

Syntax

The PKTS.STORAGE statement has the following syntax:

PKTS.STORAGE

►►PKTS.STORAGE.*stackname*=*storg*—————►►

where:

stackname

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

storg

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

Usage notes

This statement applies only if collection of packet trace data is started automatically during NetView initialization using the INIT.PKTS statement. To specify storage allocation when starting data collection manually, use the STORAGE key word on the PKTS START command. (See the *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)* for more information.)

POLICY

Purpose

The POLICY statement loads definitions in the Policy Repository when the NetView program is initialized.

Syntax

The POLICY statement has the following syntax:

►►POLICY.—*name*—=—*policy_file*—————►►

where:

name

Specifies any set of characters as defined by each policy application.

policy_file

Specifies a file name within DSIPARM that contains policy definitions.

Usage notes

- The NetView program provides the following policy files:

Table 5. NetView Policy Files

CNMSTYLE Statement	Component
POLICY.&DOMAIN = NVPOLICY	Base NetView
POLICY.NETVIEW = CNMPOLCY	Base NetView
POLICY.AON = EZLCFG01	AON
POLICY.GRAPHICS = DUILPOLCY	Graphics

- You can have one or more POLICY statements, depending on your needs.
- For more information about policy files, see
 - Chapter 4, “Policy File Definitions,” on page 409
 - IBM Tivoli NetView for z/OS Automation Guide*
 - Online help for the POLICY command

PWD

Purpose

The PWD statement can be used to define passwords for use with a data REXX function or with a REXX command in a privileged authority environment. A data REXX function is always privileged unless a member is being read by a non-privileged command such as the BROWSE command. Commands are privileged if invoked by NetView processes (as with auxInitCmd) or if an AUTBYPAS call has successfully invoked privilege. Both a data REXX function and a REXX command use the getpw() function call to obtain the password.

Syntax

The PWD statement has the following syntax:

PWD

►► PWD .—name— [.suffix] —=— password —►►

where:

name

A name to be associated with the password being defined. The *name* variable can be any sequence of eight characters. The characters can be any combination of alphanumeric characters or the three national characters: the at sign (@), the number sign (#), or the dollar sign (\$). The *name* variable does not need to be an identifier, such as an operator ID, defined to the NetView program. Use uppercase characters only.

suffix

A qualifier that can be appended at the end of *name*. This is used if more than one password is to be associated with the same *name*.

- Names qualified by .P are referred to as primary.

- Names qualified by .S are referred to as secondary.

password

The password to be associated with *name*, *name.P*, or *name.S*. The password can be any combination of 1 - 24 case sensitive alphanumeric characters or symbols, excluding blanks, equal signs, ampersands, periods, the less than symbol, and non-printable characters (00x - 3F).

Usage notes

- For passwords to be hidden, include PWD statements in member CNMSTPWD. You can use READSEC to protect CNMSTPWD from being displayed by the BROWSE command.
- The sample CNMSTPWD contains password statements that can be modified and used.

Examples

- PWD.mytask = pw1
After this, getpw('mytask') returns pw1
- PWD.DSILOG.P = pw2
After this, getpw('DSILOG','P') returns pw2
- PWD.DSILOG.S = pw3
After this, getpw('DSILOG','S') returns pw3

RCMRODMUser

Purpose

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.

Syntax

The RCMRODMUser statement has the following syntax:

RCMRODMUser

►►RCMRODMUser—=—*applid*—►►

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 and the authority to create and change RODM data.

RESTORE.CGLOBAL

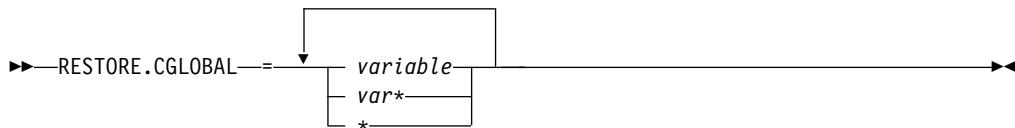
Purpose

The RESTORE.CGLOBAL statement specifies common global variable names that are to be restored during initialization of the NetView program.

Syntax

The RESTORE.CGLOBAL statement has the following syntax:

RESTORE.CGLOBAL



where:

variable

Specifies one or more saved common global variables. You can specify variables using wildcards. Separate variables with a space or a comma (,).

When you use an asterisk (*), the entire global dictionary is restored from external storage or the VSAM file. If you are restoring many variables, consider how much processor time is required for the I/O.

For more information about which variables you can specify and how to use wildcards, see the **GLOBALV RESTOREC** command.

Usage notes

- Use the RESTORE.CGLOBAL statement to ensure that the specified common global variables are restored before any message automation takes place. This allows the variables to be used in automation tables.
If common global variables are not restored until after NetView initialization (for example, using an auxInitCmd statement), they might not have a value when automation table processing references them.
- The RESTORE.CGLOBAL statement overrides the TASK.DSISVRT.INIT statement if the TASK.DSISVRT.INIT statement is set to no (N). In this case, the DSISVRT task is started and a message that indicates that the TASK.DSISVRT.INIT=N statement is ignored is issued.

Example

To restore all common global variables that begin with the letters ABC (for example ABC1, ABC2, and ABCXYZ), use the following statement:

```
RESTORE.CGLOBAL = ABC*
```

Related statements

TASK.DSISVRT.INIT

REXEC.PORT

Purpose

The REXEC.PORT statement defines the port number on which DSIREXCF waits for connection requests.

Syntax

The REXEC statement has the following syntax:

RExec.PORT



where:

port

Defines the port number. The default is 512.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE REXEC command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

REEXEC.SOCKETS

Purpose

The REXEC.SOCKETS statement specifies the maximum number of simultaneous users for the REXEC server (DSIRXEC task in the DSIREXCF member).

Syntax

The REXEC statement has the following syntax:

REEXEC.SOCKETS



where:

number

Specifies the maximum number of simultaneous users. The default is 50.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE REXEC command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

REXEC.TCPANAME

Purpose

The REXEC.TCPANAME statement defines TCP/IP address space name for the REXEC server (DSIRXEXC task in the DSIREXCF member).

Syntax

The REXEC statement has the following syntax:

REXEC.TCPANAME



where:

name

Defines the TCP/IP address space name. The default is &CNMTCPN.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE REXEC command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

REXX.CMDENV.name

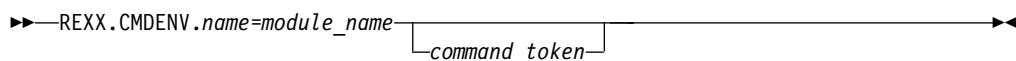
Purpose

The REXX.CMDENV.*name* statement defines NetView REXX external command environments. These REXX command environments are added to the NetView REXX environment parameters.

Syntax

The REXX.CMDENV.*name* statement has the following syntax:

REXX.CMDENV. *name*



where:

command_token

Specifies the optional 16-character token that is passed to the external command processor.

module_name

Specifies the name of the external command processor load module. For information about coding this module, see the *z/OS TSO/E REXX Reference*.

name

Indicates the 1- to 8-character external command environment name that is used in the REXX ADDRESS keyword instruction. This name must be unique in the NetView address space in which it is defined.

Usage notes

- If you specify any REXX.CMDENV.*name* statements in the CNMSTYLE (CNMSTUSR or CxxSTGEN) member, any external command environments that are specified in the DSIRXPRM module are overridden by the NetView external command environments, first by the NETVIEW and NETVASIS external command environments and then by the ones that are specified in the REXX.CMDENV statements.
- You can define up to 10 external command environments for each NetView address space. These are in addition to the NETVIEW and NETVASIS external command environments.

REXX.FUNCPKGLIST.LCL.*name*

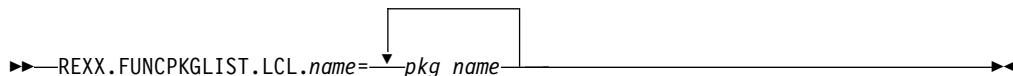
Purpose

The REXX.FUNCPKGLIST.LCL.*name* statement enables multiple NetView REXX local function packages to be defined. These REXX local function packages are added to the NetView REXX environment parameters.

Syntax

The REXX.FUNCPKGLIST.LCL.*name* statement has the following syntax:

REXX.FUNCPKGLIST.LCL.*name*



where:

name

Indicates the unique 1- to 255-character name, for example, a product name such as SAZOS, by which a group of concatenated local function packages is referred.

pkg_name

Indicates the name of a local function package to be concatenated. Up to 10 local function package names, separated by at least one blank, can be specified. For information about coding REXX function packages, see the *z/OS TSO/E REXX Reference*.

Usage notes

- If you specify any REXX.FUNCPKGLIST.LCL.*name* statements in the CNMSTYLE (CNMSTUSR or CxxSTGEN) member, all local function package names that are specified in the DSIRXPRM module are overridden by the ones that are specified in the REXX.FUNCPKGLIST.LCL.*name* statements.
- You can define up to a total of 10 package names for each NetView address space.

REXX.FUNCPKGLIST.SYS.*name*

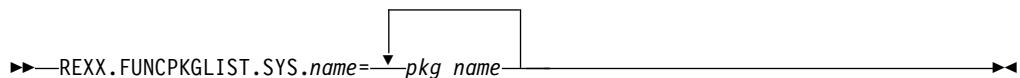
Purpose

The REXX.FUNCPKGLIST.SYS.*name* statement enables multiple NetView REXX system function packages to be defined. These REXX system function packages are added to the NetView REXX environment parameters.

Syntax

The REXX.FUNCPKGLIST.SYS.*name* statement has the following syntax:

REXX.FUNCPKGLIST.SYS.*name*



where:

name

Indicates the unique 1- to 255-character name, for example, a product name such as SAZOS, by which a group of concatenated system function packages is referred.

pkg_name

Indicates the name of a system function package to be concatenated. Up to 10 system function package names, separated by at least one blank, can be specified. For information about coding REXX function packages, see the *z/OS TSO/E REXX Reference*.

Usage notes

- If you specify any REXX.FUNCPKGLIST.SYS.*name* statements in the CNMSTYLE (CNMSTUSR or CxxSTGEN) member, all system function package names that are specified in the DSIRXPRM module are overridden by the ones that are specified in the REXX.FUNCPKGLIST.SYS.*name* statements.
 - You can define up to a total of 10 package names for each NetView address space. These are in addition to the following function package names:
 - DSIRXFPG (NetView system)
 - INGRXFPG (System Automation for z/OS), if the SA tower is enabled
- The DSIRXFPG and, if applicable, INGRXFPG package names are inserted as the first and second package names respectively.

REXX.FUNCPKGLIST.USR.*name*

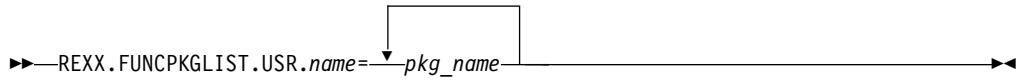
Purpose

The REXX.FUNCPKGLIST.USR.*name* statement enables multiple NetView REXX user function packages to be defined. These REXX user function packages are added to the NetView REXX environment parameters.

Syntax

The REXX.FUNCPKGLIST.USR.*name* statement has the following syntax:

REXX.FUNC PKGLIST.USR.name



where:

name

Indicates the unique 1- to 255-character name, for example, a product name such as SAZOS, by which a group of concatenated user function packages is referred.

pkg_name

Indicates the name of a user function package to be concatenated. Up to 10 user function package names, separated by at least one blank, can be specified. For information about coding REXX function packages, see the *z/OS TSO/E REXX Reference*.

Usage notes

- If you specify any REXX.FUNC PKGLIST.USR.*name* statements in the CNMSTYLE (CNMSTUSR or CxxSTGEN) member, all user function package names that are specified in the DSIRXPRM module are overridden by the ones that are specified in the REXX.FUNC PKGLIST.USR.*name* statements.
- You can define up to a 10 total of package names for each NetView address space.

RMTALIAS

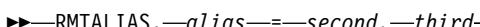
Purpose

The RMTALIAS statement specifies the RMTSYN definition that applies when the alias identifier is specified as the target domain on a RMTCMD SEND command. You can use multiple RMTALIAS statements to define a combination of networks (*netid*), domains and transports (IPv4, IPv6, or SNA).

Syntax

Syntax for the RMTALIAS statement:

RMTALIAS



where:

alias

Specifies a 1 - 8 character identifier that resolves to a particular RMTSYN name

second

Specifies the second element (1 - 8 characters) of the RMTSYN name to be used. This value corresponds to the network ID that is specified on the RMTSYN statement and might or might not be a real network ID.

third

Specifies the third element (1 - 5 characters) of the RMTSYN name to be used. This value corresponds to the domain ID specified on the RMTSYN statement and is a real domain ID.

Usage notes

The *second* position in the RMTALIAS value is not necessarily a real network ID. The ON keyword of the RMTSYN statement is used to provide the real network ID for the alias being defined by the RMTALIAS/RMTSYN statement pair.

Examples

- **Example 1: Defining an alias for the RMTCMD command over IPv6 for the CNM01 domain in the NETA network**

The following statements define the XXIPV6 alias for the CNM01 domain in the NETA network. The name of the host at which the target NetView program resides is host6.company.com, and the RMTCMD server (DSIUDST task) in the CNM01 domain is listening for connection requests on the 4022 port. When the NetView program requests resolution of the host name, an IPv6 address is requested so that the transport that is used by the RMTCMD command is TCP over IPv6.

```
RMTALIAS.XXIPV6 = XXNETA.CNM01  
RMTSYN.XXNETA.CNM01 = host6.company.com/4022 IPV6 ON NETA
```

The following RMTCMD SEND command uses the XXIPV6 alias:

```
RMTCMD SEND,DOMAIN=XXIPV6,command
```

- **Example 2: Defining an alias for the RMTCMD command over IPv4 for the CNM01 domain in the NETA network**

The following statements define the YYIP alias for the CNM01 domain in the NETA network. The name of the host at which the target NetView resides is host.company.com, and the RMTCMD server (DSIUDST task) in the CNM01 domain is listening for connection requests on the 4022 port. When the NetView program requests resolution of the host name, an IPv4 address is requested so that the transport that is used by the RMTCMD command is TCP over IP (IPv4).

```
RMTALIAS.YYIP = YYNETA.CNM01  
RMTSYN.YYNETA.CNM01 = host.company.com/4022 IPV4 ON NETA
```

The following RMTCMD SEND command uses the YYIP alias:

```
RMTCMD SEND,DOMAIN=YYIP,command
```

- **Example 3: Defining an alias for the RMTCMD command over SNA for the CNM01 domain in the NETA network**

The following statements define the ZZSNA alias for the CNM01 domain in the NETA network. The CNM01 domain is also the LU name for the target NetView application. Because the value of the RMTSYN statement indicates SNA, an SNA LU6.2 conversation with the target NetView LU is used to transport the remote command. For additional information on creating a RMTALIAS/RMTSYN pair, see the Usage Notes for the RMTSYN statement.

```
RMTALIAS.ZZSNA = ZZNETA.CNM01  
RMTSYN.ZZNETA.CNM01 = SNA ON NETA
```

The following RMTCMD SEND command uses the ZZSNA alias:

```
RMTCMD SEND,DOMAIN=ZZSNA,command
```

RMTINIT.IP

Purpose

The RMTINIT.IP statement specifies whether all TCP/IP remote operations (IPv4 and IPv6) are enabled for RMTCMD and ENDTASK processing.

For IPv6: IPv6 networking must already be enabled in the TCP/IP stack (the AF_INET6 address family is enabled) and the IPv6Env CNMSTYLE statement must be set to MIXED or ONLY.

Syntax

The RMTINIT.IP statement has the following syntax:

RMTINIT.IP



where:

YES

Specifies that TCP/IP remote operations is enabled. This is the initial setting in the CNMSTYLE member.

NO Specifies that TCP/IP remote operations is not enabled. This is the default value if you do not specify a RMTINIT.IP statement.

Usage notes

- By default, not all NetView functions have RMTCMD for TCP/IP enabled. There might be additional configuration that is required for these functions.

RMTINIT.PORT

Purpose

The RMTINIT.PORT statement specifies the port number to which DSIUDST is to bind and receives remote operations connection requests. This statement is read only if RMTINIT.IP is yes.

Syntax

The RMTINIT.PORT statement has the following syntax:

RMTINIT.PORT



where:

portnum

A number from 0 - 65535. Specifying 0 requests that TCP/IP assign a port number.

Usage Notes

If the RMTCMD command on two NetView programs are to use TCP/IP and both are using the same TCP/IP stack, specify different ports to avoid a conflict.

RMTINIT.SAFrefresh

Purpose

The RMTINIT.SAFrefresh statement specifies whether the security option for RMTCMD and ENDTASK processing can be dynamically changed from TABLE or NONE to SAF by the REFRESH command.

Syntax

The RMTINIT.SAFrefresh statement has the following syntax:

RMTINIT.SAFrefresh



where:

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.SECOPT

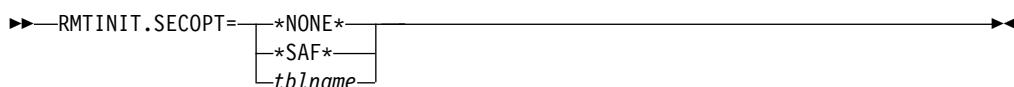
Purpose

The RMTINIT.SECOPT statement defines the type of security that you want to use for RMTCMD and ENDTASK processing. Use the security specified 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.

Syntax

The RMTINIT statement has the following syntax:

RMTINIT.SECOPT



where:

NONE

Specifies that no security check is made. Any NetView operator from any network and domain can start or end any autotask using the RMTCMD or ENDTASK command. The initial value is *NONE*.

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 - 8 character name of the table used for RMTCMD security verification. For additional information about how to build the table, see the *IBM Tivoli NetView for z/OS Security Reference*. The sample supplied with the NetView program for the RMTCMD security validation table is DSISECUR.

RMTINIT.SOCKETS

Purpose

The RMTINIT.SOCKETS statement specifies the maximum number of sockets to be available to the DSUDST task for supporting TCP/IP remote operations. This keyword is read only if RMTINIT.IP is Yes.

Syntax

The RMTINIT.SOCKETS statement has the following syntax:

RMTINIT.SOCKETS

►►RMTINIT.SOCKETS—=—*socketnum*—►►

where:

socketnum

Specifies a number from 50 - 2000.

RMTINIT.SNA

Purpose

The RMTINIT.SNA statement specifies whether SNA LU 6.2 remote operations is enabled for RMTCMD and ENDTASK processing.

Syntax

The RMTINIT.SNA statement has the following syntax:

RMTINIT.SNA

►►RMTINIT.SNA=—

NO
YES

—►►

where:

YES

Specifies that SNA LU 6.2 remote operations is enabled.

NO Specifies that SNA LU 6.2 remote operations is not enabled.

RMTINIT.TCPNAME

Purpose

The RMTINIT.TCPNAME statement specifies the name of the TCP/IP stack to be used for remote operations. This statement is read only if RMTINIT.IP is set to YES.

Syntax

The RMTINIT.TCPNAME statement has the following syntax:

RMTINIT.TCPNAME

►—RMTINIT.TCPNAME—=—*ipname*—►►

where:

ipname

Specifies the name of the TCP/IP stack to be used for remote operations.

RMTINIT.KEEPALIVE

Purpose

The RMTINIT.KEEPALIVE statement specifies the approximate number of seconds of TCP/IP communication lost for a TCP/IP-based RMTCMD connection with any other NetView domain, after which the connection is ended. This function sets the value used on the TCP_KEEPALIVE option for TCP/IP sockets. Refer to the *Communications Server: IP Programmer's Guide and Reference* manual for more information about the behavior of various TCP_KEEPALIVE values.

Syntax

The RMTINIT.KEEPALIVE statement has the following syntax:

RMTINIT.KEEPALIVE

►—RMTINIT.KEEPALIVE=*seconds*—►►

where:

seconds

A value 0 - 300. A value of 0 disables the function.

RMTSYN

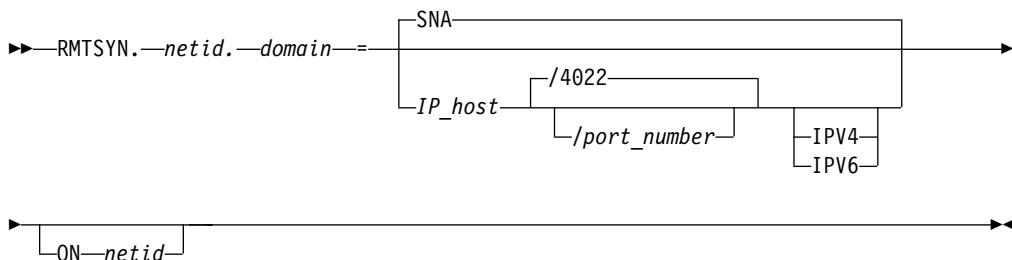
Purpose

The RMTSYN statement specifies the protocol used for communications by the PIPE ROUTE stage and by the RMTCMD and LABEL commands.

Syntax

Syntax for the RMTSYN statement:

RMTSYN



where:

netid

Specifies the 1 - 8 character name of the real network containing the *domain* or an identifier used to associate an RMTALIAS definition with the RMTSYN statement. An RMTALIAS statement is associated with this RMTSYN statement when the *second* position in the RMTALIAS value matches this identifier.

domain

Specifies the 1 - 5 character name of the remote NetView domain.

IP_host

Specifies a standard IP host name or address for the remote NetView domain. When communicating using an IP transport, the IP host can be followed by a port number. IP host names are resolved to IP addresses after the main part of initialization.

port_number

Specifies the port number. A slash (/) separates the *port_number* from *IP_host*. The default port number is 4022.

IPV4

Specifies to use the IPV4 protocol to communicate with the specified *domain*. Use the IPV4 keyword with *IP_host* when it is a host name.

IPV6

Specifies to use the IPV6 protocol to communicate with the specified *domain*. Use the IPV6 keyword with *IP_host* when it is a host name.

ON netid

Specifies the 1 - 8 character name of the network containing the domain. Use the ON *netid* specification when an RMTSYN statement is associated with an RMTALIAS statement.

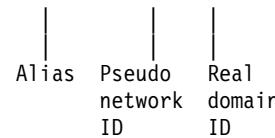
SNA

Specifies to use SNA LU 6.2 to communicate with the specified *domain*. This is the default.

Usage notes

- If the IPV4 or IPV6 keyword is coded in an RMTSYN value and if the target host is identified by an IP address, then the IPV4 or IPV6 keyword (whichever is coded) is ignored by RMTCMD command processing.
- The network ID that is specified with the ON keyword is used as the network ID for a target NetView domain under the following conditions:
 - NETID=* is specified or set as the default value for a RMTCMD SEND command.
 - RMTSYN statement is part of a RMTALIAS/RMTSYN statement pair that is defining an alias for a target NetView domain.
- If the RMTSYN definition is part of a RMTALIAS/RMTSYN statement pair used to define an alias, the RMTSYN statement cannot be used to resolve the value of the DOMAIN keyword of RMTCMD unless the pseudo-network-ID portion of the RMTALIAS and RMTSYN statements matches the real network ID of the NetView domain to be targeted. Consider this example:
 - Define an alias, NTVGGV6, for remote domain NTVGG, specifying an IPv6 network transport, a host name, port number 40220, and real network ID USIBMNT

```
RMTALIAS.NTVGGV6 = XXV6.NTVGG
```



```
RMTSYN.XXV6.NTVGG = your.host.com/40220 IPV6 ON USIBMNT
```



With these definitions, the NTVGGV6 alias may be used to send a command to NetView domain NTVGG in network USIBMNT like this:

```
RMTCMD SEND DOMAIN=NTVGGV6 OPERID=OPER1 AUTOTBL STATUS
```

If you attempt to send a command from the current NetView domain, presumed to also be in network USIBMNT in this example, to NetView domain NTVGG like this:

```
RMTCMD SEND DOMAIN=NTVGG OPERID=OPER1 AUTOTBL STATUS
```

The RMTSYN statement cannot be used to resolve NetView domain NTVGG in network USIBMNT to the applicable IP address, port, and network transport, because the network ID portion of the RMTSYN statement, XXV6, does not match the real network ID, USIBMNT.

Examples

- An example RMTSYN statement follows using an IP host name:

```
RMTSYN.NETA.CNM01 = host6.company.com/4022 // resolve later
```

Because this RMTSYN statement does not include a transport keyword, the IP (IPv4 or IPv6) address is determined based on the setting of the IPv6Env statement:

- NONE: IPv4 address is used, if available
- ONLY: IPv6 address is used, if available
- MIXED: first address returned is used (either IPv4 or IPv6)
- An example RMTSYN statement follows using an IP address and a default port:
RMTSYN.NETB.CNM03 = 10.67.50.22 // IP address & default port
- An example RMTSYN statement follows for a session using LU 6.2:
RMTSYN.NETX.NTVAA = SNA // use LU6.2 for this domain
- An example RMTSYN statement follows for a session using IPv6:
RMTSYN.NETX.NTVXX = 2002:92A:111:501:10:10:163:7/4022

For additional examples, see the RMTALIAS statement.

RODMname

Purpose

The RODMname statement specifies the RODM name as a global variable.

Syntax

The RODMname statement has the following syntax:

RODMname

►►—RODMname—=—*RODM_name*—►►

where:

RODM_name

Indicates a 1 - 5 character RODM name.

Usage notes

- This statement is ignored if you are not using RODM.
- If you set the &CNMRODM system symbolic variable in the IEASYMxx member in the SYS1.PARMLIB data set, do not modify the RODMname statement.
- If you are using GMFHS, specify the same RODM name in the DUINIT member as you specify on this statement.

ROUTECD

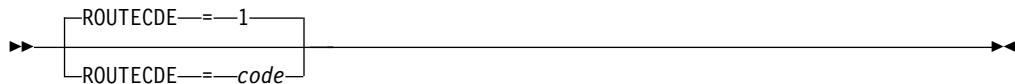
Purpose

The ROUTECDE statement specifies a single routing code to be assigned to NetView messages issued with the WTO and WTOR commands or macros, when those commands or macros do not specify a routing code.

Syntax

The ROUTECDE statement has the following syntax:

ROUTECDE



where:

code

The assigned routing code. The default value is 1.

Usage notes

- Any messages issued before the completion of CNMSTYLE processing use a default ROUTECDE of 1. This field also results in the setting of the common global variable ROUTECODE.

RRD

Purpose

The RRD statement controls the usage of the START DOMAIN command for the operators and autotasks with profiles that specify a control value of GLOBAL (CTL=GLOBAL). For these tasks, the START DOMAIN command provides access to a remote NetView program if either the *domainid* is specifically listed on an RRD statement, or an RRD statement with an asterisk has allowed for a sufficient count of unspecified domains.

Syntax

The RRD statement has the following syntax:



where:

domainid

Indicates the 1 - 5 character NetView program identifier of the domain with which the operators with CTL= GLOBAL can start an NNT cross-domain session. The value of *domainid* must match the NetView program domain identifier in the other domain. The value specified in this case is always an asterisk.

For interconnected networks, all NetView systems in session with each other must have unique identifiers that cannot be alias names.

- * When an asterisk (*) is coded instead of *domainid*, it indicates that a NetView operator or autotask with CTL=GLOBAL can start an NNT cross-domain session by issuing the START DOMAIN command to a *domainid* that is not defined by any RRD statement.

nn A numeric value of 0 - 99. This is the number of concurrent NNT sessions each NetView operator with CTL=GLOBAL can have.

Usage notes

- The RRD statements are not affected by a RESTYLE command

- Conversations with distributed autotasks that are started with the RMTCMD command are not affected by the RRD statement.
- To restrict a specific operator from using the START DOMAIN command with any *domainid* value, use security restrictions for the combination of the DOMAIN keyword and the *domainid* value.
- If no RRD.*=nn statement is coded, or if the value that is specified for *nn* is 0, then a NetView operator with CTL=GLOBAL can start only NNT sessions for the domains that are defined on an RRD.domain=* statement; no dynamic NNT sessions can be started.
- If RRD.* is specified, but no value or an invalid value is specified for *nn*, then a default value of 20 is used.

RSH.PORT

Purpose

The RSH.PORT statement defines the port number on which DSIRSH waits for connection requests. This statement is used for the RSH server (DSIRSH task in the DSIRSHCF member).

Syntax

The RSH.PORT statement has the following syntax:

RSH.PORT



where:

port

Specifies the port number. The default is 514.

Usage notes

- RSH uses the value on the TCPname statement for the value of the TCP/IP address space name.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE RSH command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

RSH.SOCKETS

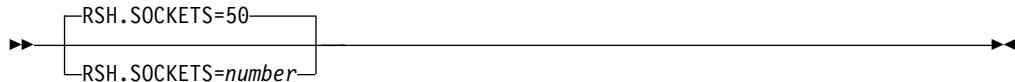
Purpose

The RSH.SOCKETS statement specifies the maximum number of simultaneous users. This statement is used for the RSH server (DSIRSH task in the DSIRSHCF member).

Syntax

The RSH statement has the following syntax:

RSH.SOCKETS



where:

SOCKETS = number

Specifies the number of simultaneous users. The default is 50.

Usage notes

- RSH uses the value on the TCPname statement for the value of the TCP/IP address space name.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE RSH command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

RTNDEF.BASE.COLLECTOR.APPL

Purpose

The RTNDEF.BASE.COLLECTOR.APPL statement specifies the data collector for application information.

Syntax

The RTNDEF.BASE.COLLECTOR.APPL statement has the following syntax:

```
►-- RTNDEF.BASE.COLLECTOR.APPL == CNMEDCAP --►
```

RTNDEF.BASE.COLLECTOR.STACK

Purpose

The RTNDEF.BASE.COLLECTOR.STACK statement specifies the data collector for stack information. SNMP commands are issued from the CNMEDCST command list to gather stack information.

Syntax

The RTNDEF.BASE.COLLECTOR.STACK statement has the following syntax:

```
►-- RTNDEF.BASE.COLLECTOR.STACK == CNMEDCST --►
```

Usage notes

To disable the stack data collector function, specify *NONE* on the RTNDEF.BASE.COLLECTOR.STACK statement:

RTNDEF.BASE.COLLECTOR.STACK = *NONE*

RTNDEF.BASE.COLLECTOR.SYSTEM

Purpose

The RTNDEF.BASE.COLLECTOR.SYSTEM statement specifies the data collector for system information.

Syntax

The RTNDEF.BASE.COLLECTOR.SYSTEM statement has the following syntax:

►—RTNDEF.BASE.COLLECTOR.SYSTEM—=—CNMEDCSY————►

RTNDEF.BASE.HB.APPL

Purpose

The RTNDEF.BASE.HB.APPL statement specifies the time interval to be used when defining checks for connectivity of remote NetView applications that have been discovered.

Syntax

The RTNDEF.BASE.HB.APPL statement has the following syntax:

RTNDEF.BASE.HB.APPL

►—RTNDEF.BASE.HB.APPL=00:03:00————►
 |————RTNDEF.BASE.HB.APPL=hh:mm:ss————|————►

where:

hh:mm:ss

Indicates the interval in hours (24-hour clock), minutes, and seconds for a check on connectivity. The initial default value is 00:03:00 (3 minutes).

Usage notes

If connectivity is lost, all resources reported by the NetView application have a status of UNKNOWN.

RTNDEF.BASE.MEMBER

Purpose

The RTNDEF.BASE.MEMBER statement specifies the member name where discovered data is stored.

Syntax

The RTNDEF.BASE.MEMBER statement has the following syntax:

RTNDEF.BASE.MEMBER



where:

member_name

Specifies the member name. The initial default member name is CNMALLDA.

Usage notes

The *member_name* is an INSTORE member only.

RTNDEF.BASE.PROCESSOR.APPL

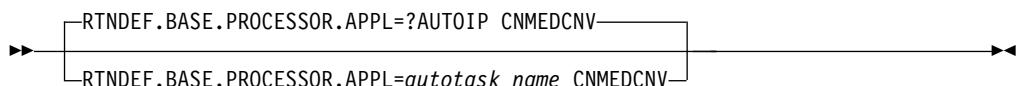
Purpose

The RTNDEF.BASE.PROCESSOR.APPL statement specifies the routine that is called for application events and the autotask to which the routine is routed for automation.

Syntax

The RTNDEF.BASE.PROCESSOR.APPL statement has the following syntax:

RTNDEF.BASE.PROCESSOR.APPL



where:

autotask_name

Autotask to which stack events are routed. The initial setting is ?AUTOIP.

RTNDEF.BASE.PROCESSOR.STACK

Purpose

The RTNDEF.BASE.PROCESSOR.STACK statement specifies the entity processor for discovered stacks.

Syntax

The RTNDEF.BASE.PROCESSOR.STACK statement has the following syntax:

RTNDEF.BASE.PROCESSOR.STACK



where:

autotask_name

Task under which the entity processor runs. ?AUTOIP is the default value.

RTNDEF.BASE.PROCESSOR.SYSTEM

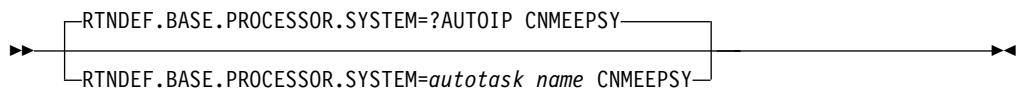
Purpose

The RTNDEF.BASE.PROCESSOR.SYSTEM statement specifies the entity processor for discovered systems.

Syntax

The RTNDEF.BASE.PROCESSOR.SYSTEM statement has the following syntax:

RTNDEF.BASE.PROCESSOR.SYSTEM



where:

autotask_name

Task under which the entity processor runs. ?AUTOIP is the default value.

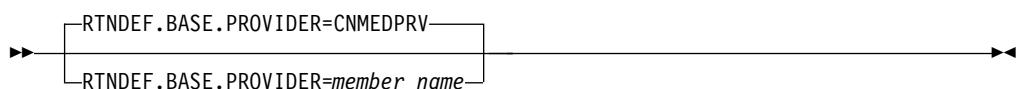
RTNDEF.BASE.PROVIDER

Purpose

The RTNDEF.BASE.PROVIDER statement specifies the routine to use to query the discovered resource data.

Syntax

The RTNDEF.BASE.PROVIDER statement has the following syntax:



where:

member_name

Specifies the member name. The initial default value is member CNMEDPRV.

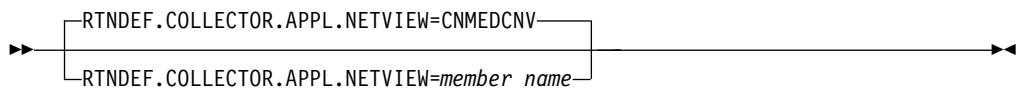
RTNDEF.COLLECTOR.APPL.NETVIEW

Purpose

The RTNDEF.COLLECTOR.APPL.NETVIEW statement specifies the data collector for application information.

Syntax

The RTNDEF.COLLECTOR.APPL.NETVIEW statement has the following syntax:



where:

member_name

Specifies the member name. The initial default value is member CNMEDCNV.

RTNDEF.PROCESSOR.STACK

Purpose

The RTNDEF.PROCESSOR.STACK statement specifies the routine that is called for stack events and the autotask to which the routine is routed for automation.

Syntax

The RTNDEF.PROCESSOR.STACK statement has the following syntax:



where:

autotask

Autotask to which stack events are routed. For AON, the initial setting is ?AUTOIP.

routine

Name of the routine to be called for stack events.

Examples

- In the following examples, stack events are processed by the autotask defined by AUTOIP. FFXESTCK is the routine called to process stack events for AON.TCP or IPMGT.

(AON.TCP)RTNDEF.PROCESSOR.STACK.AON = ?AUTOIP FFXESTCK
(IPMGT)RTNDEF.PROCESSOR.STACK.AON = ?AUTOIP FFXESTCK

RTT.PORT

Purpose

The RTT.PORT statement defines the port number on which the alert receiver (DSIRTTR task in the DSIRTTD member) waits for connection requests.

Syntax

The RTT.PORT statement has the following syntax:

RTT.PORT



where:

number

Defines the port number. The default is 4021.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE RTT command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

RTT.SOCKETS

Purpose

The RTT.SOCKETS statement specifies the maximum number of simultaneous requests. This statement is used with the alert receiver (DSIRTTR task in the DSIRTTTD member).

Syntax

The RTT.SOCKETS statement has the following syntax:

RTT.SOCKETS



where:

number

Specifies the maximum number of simultaneous requests. The default is 50.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE RTT command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

RTT.TCPANAME

Purpose

The RTT.TCPANAME statement defines the TCP/IP address space name. This statement is used for the alert receiver (DSIRTTR task in the DSIRTTTD member).

Syntax

The RTT.TCPANAME statement has the following syntax:

RTT.TCPANAME

►—RTT.TCPANAME=*name*————→

where:

name

Defines the TCP/IP address space name.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE RTT command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

SECOPTS.AUTHCHK

Purpose

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 *IBM Tivoli NetView for z/OS Security Reference*.

Syntax

The SECOPTS.AUTHCHK statement has the following syntax:

►—

SECOPTS.AUTHCHK=SOURCEID
SECOPTS.AUTHCHK=TARGETID

————→

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.

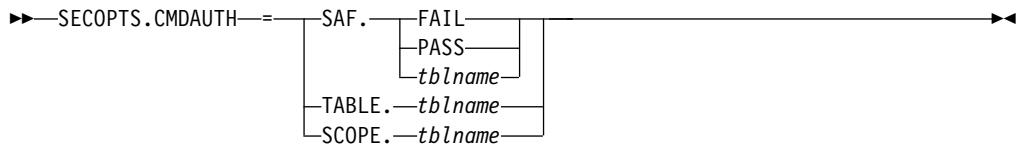
SECOPTS.CMDAUTH

Purpose

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 *IBM Tivoli NetView for z/OS Security Reference*.

Syntax

The SECOPTS.CMDAUTH statement has the following syntax:



where:

SAF.option

Specifies that the NetView program performs command authorization checking using an SAF security product. Define the commands that you want to protect 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 *IBM 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. Protect these commands using the backup command authorization table.

Include one of the following options when you specify SAF:

FAIL Specifies that users are not allowed to issue any commands when the SAF product cannot make a security decision.

PASS Specifies that users are 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.
- No resource name is defined in the NETCMDS class which protects or authorizes this command.
- The NETCMDS class is not active.
- The security product is not active.

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 about how to build the table, see the *IBM 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 about how to build the table, see the *IBM 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 a CMDDEF 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 *IBM Tivoli NetView for z/OS Security Reference*.
- Scope of commands is supported in migration mode only. Any CMDCLASS, KEYCLASS, or VALCLASS statements in DSICMD, in addition to 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 6 shows the relationships between CMDAUTH keywords and SECTOPS.AUTHCHK specifications.

Table 6. Interrelationships between SECOPTS.CMDAUTH Keyword Values

IF CMDAUTH=	TABLE	SAF. <i>tblname</i>	SAFPASS FAIL	SECOPTS.AUTHCHK=	
				TARGETID	SOURCEID
TABLE	Required	Ignored	Ignored	Is valid	Default
SAF	Ignored	Is valid	Is valid, defaults to PASS, and is ignored if <i>tblname</i> is specified	Is valid	Default

- Table 7 shows how the SECOPTS.CMDAUTH and related options are used to specify how operator command authority checking is to be performed.

Table 7. Protecting Commands Processed in the NetView Program

Keyword	Restrictions	Effect
SECOPTS.CMDAUTH = TABLE		Command authorization is based on the specified table.
SECOPTS.CMDAUTH = SAF	<p>MVS RACF Version 2 or equivalent SAF security product is required.</p> <p>Cannot be specified if SECOPTS.OPERSEC is MINIMAL, NETVPW, or SAFPW.</p>	<p>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.</p>

Table 7. Protecting Commands Processed in the NetView Program (continued)

Keyword	Restrictions	Effect
SECOPTS.CMDAUTH = SAF. <i>tblname</i>		<p>Specifies the backup table to be used for immediate commands and when the SAF product cannot make a security decision. This can occur when:</p> <ul style="list-style-type: none"> • No resource name is 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 = SAF.PASS FAIL	Defaults to PASS	<p>Identifies whether to PASS or FAIL command authority checking if the SAF product can reach no decision.</p> <ul style="list-style-type: none"> • For more information, refer to <i>IBM Tivoli NetView for z/OS Security Reference</i>.

SECOPTS.OPERSEC

Purpose

The SECOPTS.OPERSEC statement defines the method used to allow users to log on to the NetView program.

Syntax

The SECOPTS.OPERSEC statement has the following syntax:



where:

MINIMAL

Specifies that NetView operators are defined by a list of operator identifiers in DSIOPF. No password validation is done. 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.

NETV PW

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.

Do not specify SECOPTS.CMDAUTH=SAF or SECOPTS.OPSPAN=SAF when SECOPTS.OPERSEC=NETVPW.

SAFPW

Specifies that operator identification and password or password phrase

checking is performed using an SAF security product. The operator identifier must also be defined in DSIOPF, unless a default logon profile is defined by using the DEFAULTS command or the DEFAULTS.LogProf statement in the CNMSTYLE member. Other attributes given to the operator at logon are taken from the specified profile for the operator in DSIPRF.

Security access checks are verified against the authority of the NetView program, and not the authority of the operator, when an operator tries to access the following items:

- Data set that is protected in the DATASET class of an SAF product
- An MVS system command that is protected in the OPERCMDS class of an SAF product

Do not specify SECOPTS.CMDAUTH=SAF or SECOPTS.OPSPAN=SAF when SECOPTS.OPERSEC=SAFPW. If SECOPTS.OPSPAN=SAF is specified, SECOPTS.OPSPAN=NETV is used.

SAFCHECK

Specifies that operator identification and password or password phrase checking is performed using an SAF security product. The operator identifier must also be defined in DSIOPF, unless a default logon profile is defined by using the DEFAULTS command or the DEFAULTS.LogProf statement in the CNMSTYLE member. Other attributes given to the operator at logon are taken from the specified profile for the operator in DSIPRF. This includes span of control information, unless SECOPTS.OPSPAN=SAF is specified or OPSPAN=SAF is specified on the REFRESH command. In this case, span of control information is taken from the NETSPAN class of the SAF product.

Security access checks are checked against the authority of the operator that 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.

This specification provides the same function as the 'SAF CHECK AT TASK LEVEL' byte in the NetView constants module DSICTMOD 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 DSICTMOD byte is set to X'01' an error message is issued and initialization continues.

SAFDEF

Specifies that operator identification and password or password phrase 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 user profile for the operator in the SAF product. For more information, refer to *IBM 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 you are using an SAF security product to perform operator identification and password or password phrase checking, you can log on to the NetView program using a PassTicket rather than a password or password phrase. To use a PassTicket, use an SAF product which supports PassTickets, such as the RACF Network Security Program/Secure Logon Coordinator product (NetSP/SLC V1.2).
- Table 8 shows the relationships between SECOPTS.OPERSEC, SECOPTS.CMDAUTH, SECOPTS.OPSPAN, and SECOPTS.SPANAUTH specifications.

Table 8. Interrelationships between SECOPTS Statements

SECOPTS. OPERSEC=	CMDAUTH		OPSPAN		SPANAUTH	
	TABLE	SAF	NETV	SAF	*NONE*	TABLE
MINIMAL	Not valid	Not valid	Not valid	Not valid	Not valid	Not valid
NETVPW	Is valid		Default	Forced to NETV	Default	Is valid
SAFPW	Is valid		Default	Forced to NETV	Default	Is valid
SAFCHECK	Is valid	Is valid	Default	Is valid	Default	Is valid
SAFDEF	Is valid	Is valid	Forced to SAF	Default	Default	Is valid

- Table 9 shows how the SECOPTS.OPERSEC and SECOPTS.OPSPAN statements are used to specify how operator verification and authority checking is to be performed.

Table 9. Defining and Verifying Operator Authority

Keyword	Related Defaults	Restrictions	Effect
SECOPTS.OPERSEC= MINIMAL		<ul style="list-style-type: none"> SECOPTS.SPANAUTH is ignored. SECOPTS.CMDAUTH is ignored. SECOPTS.OPSPAN is ignored. Cannot use REFRESH to change OPERSEC. 	<ul style="list-style-type: none"> Logon profile ignored Logon operands ignored No password validation Operator must be defined in DSIOPF
SECOPTS.OPERSEC= NETVPW	<ul style="list-style-type: none"> SECOPTS.OPSPAN default and only choice is NETV. 	<ul style="list-style-type: none"> Not valid when SECOPTS.CMDAUTH= SAF Not valid when SECOPTS.OPSPAN= SAF 	<ul style="list-style-type: none"> Password validated from DSIOPF Operator must be defined in DSIOPF Profile read from DSIPRF

Table 9. Defining and Verifying Operator Authority (continued)

Keyword	Related Defaults	Restrictions	Effect
SECOPTS.OPERSEC=SAFPW	<ul style="list-style-type: none"> SECOPTS.OPSPAN default and only choice is NETV. The DSICTMOD byte for task-level checking is ignored. 	<ul style="list-style-type: none"> Not valid when SECOPTS.CMDAUTH= SAF Not valid when SECOPTS.OPSPAN= SAF 	<ul style="list-style-type: none"> Password or password phrase verification using SAF product Operator must be defined in DSIOPF Profile read from DSIPRF NetView startup procedure name is used for any SAF calls made for NetView operators, such as to the DATASET class
SECTOPS.OPERSEC=SAFCHECK	<ul style="list-style-type: none"> SECOPTS.OPSPAN default is NETV. 		<ul style="list-style-type: none"> Password or password phrase verification using SAF product Operator must be defined in DSIOPF Profile read from DSIPRF NetView task user IDs are used for any SAF calls for NetView operators, such as to the DATASET class Replaces the use of DSICTMOD byte setting to indicate SAF checking at the task level
SECOPTS.OPERSEC=SAFDEF	<ul style="list-style-type: none"> SECOPTS.OPSPAN default and only choice is SAF. 	<ul style="list-style-type: none"> RACF Version 2 or equivalent SAF security product is required Cannot specify OPSPAN= NETV 	<ul style="list-style-type: none"> Password or password phrase verification using SAF product Operator logon authority using RACF APPL class Operator attributes defined in NETVIEW segment of SAF product NetView task user IDs are used for any SAF calls for NetView operators, such as to the DATASET class.
SECOPTS.OPSPAN= NETV		<ul style="list-style-type: none"> If SECOPTS.OPERSEC= SAFDEF, then SECOPTS.OPSPAN= NETV is not valid. 	<ul style="list-style-type: none"> NetView program authorizes operator to start spans through the SPAN and ISPLAN statements. Operator must be defined in DSIOPF. Profile read from DSIPRF.

Table 9. Defining and Verifying Operator Authority (continued)

Keyword	Related Defaults	Restrictions	Effect
SECOPTS.OPSPAN= SAF	If SECOPTS.OPERSEC= SAFDEF, OPSPAN default is SAF.	<ul style="list-style-type: none"> 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 	<ul style="list-style-type: none"> The NETSPAN class in the security product authorizes the operator to start spans. Access level determines commands operator can issue against resources in a span. <p>• For more information, refer to <i>IBM Tivoli NetView for z/OS Security Reference</i>.</p>

SECOPTS.OPSPAN

Purpose

The SECOPTS.OPSPAN statement defines the method for determining the authority of an operator to start spans of control.

Syntax

The SECOPTS.OPSPAN statement has the following syntax:

```

    ➤—SECOPTS.OPSPAN—=—[NETV]—
                                |—————[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 that 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 *IBM 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 8 on page 236.
- 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.RMTAUTH

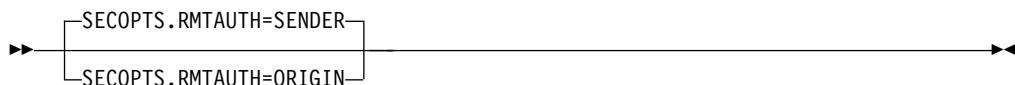
Purpose

Use the SECOPTS.RMTAUTH statement to specify the method to be used to determine which operator ID is used as the remote operator for security checks performed on RMTCMD and ENDTASK requests. This is used only for incoming requests and is not used for command security checking.

Use the SECOPTS.RMTAUTH statement together with the RMTINIT.SECOPT statement to restrict a remote operator from running commands on this NetView program using a RMTCMD or ENDTASK command.

Syntax

The SECOPTS.RMTAUTH statement has the following syntax:



where:

ORIGIN

Specifies to use the operator ID that originated the request.

SENDER

Specifies to use the operator ID that sent the RMTCMD or ENDTASK request. This is the default value.

Usage notes

- The operator ID used for comparison is the fully-qualified operator ID that includes the NetView domain and the network ID.
- You can use the REFRESH command to dynamically change the setting of the SECOPTS.RMTAUTH statement.

Examples

- The OPER1 operator ID on the CNM01 domain (NETA network) issues the following command:

```
EXCMD AUTO1 RMTCMD SEND LU=CNM02,command
```

The following CNMSTYLE statement is in effect on the CNM02 domain:

```
SECOPTS.RMTAUTH=ORIGIN
```

In this case, the NetView program on the CNM02 domain verifies that the OPER1 operator ID from the CNM01 domain (NETA network) has the authority to send the RMTCMD request.

- If instead, the following CNMSTYLE statement is in effect on the CNM02 domain:

```
SECOPTS.RMTAUTH=SENDER
```

The NetView program on the CNM02 domain verifies that the AUTO1 operator ID has the authority to send the *command*.

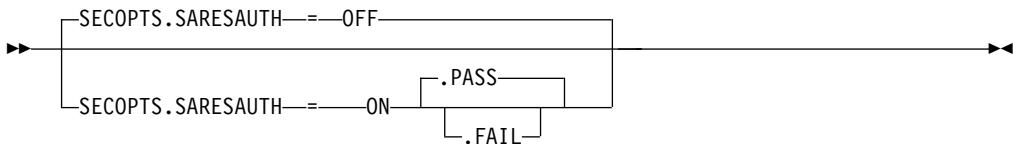
SECOPTS.SARESAUTH

Purpose

The SECOPTS.SARESAUTH statement specifies whether to check resource level security for System Automation for z/OS resources.

Syntax

The SECOPTS.SARESAUTH statement has the following syntax:



where:

OFF

Specifies that System Automation for z/OS resource level security checking is not active. No authorization checks are made. OFF is the default if the SECOPTS.SARESAUTH statement is not specified.

ON.*action*

Specifies that System Automation resource level security checking is active.

Security authorization is made by using profiles that are specified for specific System Automation for z/OS resources (for example, subsystems or groups). For information about resource level security checking, see the System Automation for z/OS library.

You can specify the *action* to be taken when the following conditions occur:

- The SAF product (such as RACF) is not active
- A resource profile that matches the resource that is being checked is not found

FAIL Specifies that the resource security request is denied.

PASS Specifies that the resource security request is granted.

Usage notes

- SARESAUT is a synonym for SARESAUTH.
- The SECOPTS.SARESAUTH=ON statement is equivalent to the SECOPTS.SARESAUTH=ON.PASS statement.

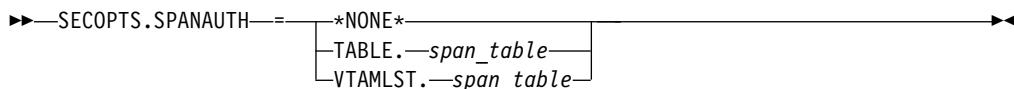
SECOPTS.SPANAUTH

Purpose

The SECOPTS.SPANAUTH statement specifies the location of the span definitions for resources and views.

Syntax

The SECOPTS.SPANAUTH statement has the following syntax:



where:

NONE

No spans are defined. When this is specified, an operator whose profile specifies CTL=SPECIFIC in the AUTH statement is not able to access any resources, because spans cannot be matched to any spans in a span table.

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 *IBM 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 SECMIGR 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 notes

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.

SECOPTS.SPANCHK

Purpose

The SECOPTS.SPANCHK statement specifies the operator ID that is used to define span checking. For specific information about how the SOURCEID and TARGETID are determined, see *IBM Tivoli NetView for z/OS Security Reference*.

Syntax

The SECOPTS.SPANCHK statement has the following syntax:



where:

SOURCEID

Specifies to check the authority of the original issuer or the ID closest to the original issuer of this VTAM command. The command issuer must be logged on when the VTAM command is run, except for VTAM commands that are issued from a system console where no logon has been performed. These commands have a source ID of *BYPASS* and are assumed to be fully authorized.

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.

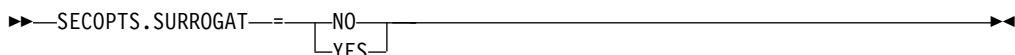
SECOPTS.SURROGAT

Purpose

The SECOPTS.SURROGAT statement specifies whether the NetView operator ID is checked to determine if it is a surrogate of a TSO user ID.

Syntax

The SECOPTS.SURROGAT statement has the following syntax:



where:

NO Surrogate checking is not performed.

YES

Surrogate checking is performed if SECOPTS.OPERSEC is set to SAFCHECK in the CNMSTYLE member or OPERSEC=SAFCHECK is specified on the REFRESH command. This check is performed during START TSOSERV, STOP TSOSERV, and TSO PIPE stage processing.

SECOPTS.WEBAUTH

Purpose

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.

Syntax

The SECOPTS.WEBAUTH statement has the following syntax:



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.

SSI.PPI

Purpose

The SSI.PPI statement specifies whether to start the Program-to-Program Interface (PPI) when the SSI is started.

Syntax

The SSI statement has the following syntax:

SSI.PPI



where:

DEFAULT

Allows the CNMCSSIR PPIOPT value to be the default.

No

Do not start the PPI.

YES

Start the PPI.

SSI.ProcString

Purpose

The SSI.ProcString statement specifies whether to start the NetView Subsystem Interface (SSI) and which parameters to use.

Syntax

The SSI.ProcString statement has the following syntax:

SSI.ProcString

►►SSI.ProcString=proc_name►►
 └─*NONE*─┘

where:

proc_name

Specifies the z/OS procedure name and parameters. The maximum length is 110 characters.

NONE

Specifies that the CNMCSSIR task does not start the address space.

Usage notes

- You can specify other parameters of the SSI start procedure through keywords in the SSI.ProcString or by modifying the SSI start procedure.

SSI.ReviseTable

Purpose

The SSI.ReviseTable statement specifies the revision table member name.

Syntax

The SSI.ReviseTable statement has the following syntax:

SSI.ReviseTable

►►SSI.ReviseTable=member_name►►
 └─*NONE*─┘

where:

member_name

Specifies the revision table member name to be loaded.

NONE

Specifies that no revision table is to be loaded.

Usage notes

- When the SSI address space is started, the CNMCSSIR task can load a revision table. A revision table can have command revision statements, message revision statements, or both types of statements. Both command and message revision statements can be included by using the %INCLUDE statement. The type of tables that are loaded depends on the statement types in the specified member (and any included statements). If either a command revision table or a message revision table is already active in the SSI address space when the CNMCSSIR task is started, the table is not loaded. When the SSI address space is started, the CNMCSSIR task performs the following actions:

- Any revision tables that were active when the NetView program was previously running are reactivated. This includes both the Message Revision Table and the Command Revision Table.
- If no revision tables were active when the NetView program was previously running, the table name specified on SSI.ReviseTable (if any) is activated.
- Any revision variables defined when the SSI address space was last terminated are reinstated.

For more information, see the REVISE command in the NetView online help or *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)*.

styleMsg

Purpose

The styleMsg statement specifies the message to be issued when CNMSTYLE processing begins.

Syntax

The styleMsg statement has the following syntax:

styleMsg

►—styleMsg—=—msg_text—►

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. See “styleMsg.wrap” for a means of specifying how 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 the CNMSTYLE member for coding examples.
- Variables are resolved in the message text, even when they are defined in statements that follow this statement in the CNMSTYLE member.

styleMsg.wrap

Purpose

The styleMsg.wrap statement specifies how a system console message statement is continued (wrapped) over more than one line.

Syntax

The styleMsg.wrap statement has the following syntax:

styleMsg.wrap

►►—styleMsg.wrap—=—*number*—►►

where:

number

Specifies the length at which message text on the system console is continued to the another line. Words are not broken.

Usage notes

- Consider using this statement to limit the length of message text sent to the system console. These messages are displayed at the system console and, optionally wrapped to aid readability. By default, messages are broken into new lines after 54 characters. You can change the character count to another value. For example, a specification of

styleMsg.wrap=30

results in this console message wrapping to the next line after a count of 30 letter positions is reached:

```
styleMsg = NetView initialization style  
sheet processing has begun.
```

stylevar

Purpose

The stylevar statement defines variables that can be used within the CNMSTYLE member.

Syntax

The stylevar statement has the following syntax:

stylevar

►►—stylevar.—*variable_name*—=—*value*—►►

where:

variable_name

Specifies the name of the variable to be used. The variable name can be 1 - 15 characters.

value

Specifies the value to be substituted when the *variable_name* is encountered. This value must have fewer than 255 characters.

Usage notes

- CNMSTYLE variables are valid only within CNMSTYLE member processing.

- The variables CNMNETID, CNMRODM, CNMTCPN, DOMAIN, NV2I, and SLASH are reserved for use by the NetView program.
- CNMSTYLE variables cannot be split across lines. Ensure that all the characters that are part of a CNMSTYLE variable, including the beginning ampersand (&) and the ending period (.), are on the same line.
- wherever two forward slashes (//) are needed together, such as in a URL, you must use the SLASH CNMSTYLE variable for one of the slashes.
- You can define a variable and use it anywhere within the CNMSTYLE member. To use a variable, specify &*variable_name*. For example, the following statement defines a variable named NVFORZOS and then shows how it can be used in another CNMSTYLE statement:


```
stylevar.NVFORZOS=netview-zos
```

```
website.url.NETVIEW.zb=http:&SLASH./www.ibm.com/software/tivoli/products/&NVFORZOS.
```
- System symbolic variables are resolved before stylevar symbolic variables.
- The construction &&*variable_name* is not supported for stylevar symbolic variables.

SuppChar

Purpose

The SuppChar statement specifies the suppression character to be used to prevent a command from being displayed on the terminal screen, hardcopy log, or NetView log.

Syntax

The SuppChar statement has the following syntax:

SuppChar

►—SuppChar—=—*character*—►

where:

character

Is the single character used as the suppression character.

Do not use any of the following characters as the suppression character:

- Ampersand (&)
- Asterisk (*)
- Blank
- Comma (,)
- Equal sign (=)
- Minus (-)
- Percent (%)
- Period (.)
- Forward slash (/)
- Alphabetic characters A - Z
- Any character that might start a command

The question mark (?) is the default.

Usage notes

To prevent operators from suppressing command logging, specify *NONE* for the suppression character:

```
SuppChar = *NONE*
```

The REXX suppchar() function continues to work in REXX procedures.

TAMEL.CONV.sysdef

Purpose

The TAMEL.CONV.sysdef statement defines the IP server to which a NETCONV session with the NetView host is to be started. This is used with CNMTAMEL (member DUILFPMEM).

Syntax

The TAMEL.CONV.sysdef statement has the following syntax:

TAMEL.CONV.sysdef

```
►—TAMEL.CONV.sysdef= ipid [ /port ]  
          SNA
```

where:

sysdef

Specify the IP or SNA server to which a NETCONV session with the NetView host is to be started.

For an SNA server, this is the LU name. An example follows:

```
TAMEL.CONV.NT1AI100 = SNA
```

ipid/port

Specifies the host name or IP address and, if specified, the port number. The port number is not required if it is the same value as that specified by the TAMEL.PORT statement. The number of host names or IP addresses that are specified cannot exceed the value specified on the TAMEL.SOCKETS statement. The maximum number of IP sessions is 2000. The following are example statements that include both IPv4 and IPv6 addresses:

- IPv4 address with a port number of 4020

```
TAMEL.CONV.IP1 = 111.22.33.144/4020 // predefined IP netconv session
```

- IPv6 address:

```
TAMEL.CONV.IP2 = 1080::8:800:200C:417A // predefined IP netconv session
```

- host name:

```
TAMEL.CONV.IP3 = xxxxx.yyyyy.com // predefined IP netconv session
```

SNA

Specifies the SNA server. An example statement follows:

```
TAMEL.CONV.NT1AI100 = SNA // predefined LU6.2 netconv session
```

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

TAMEL.PORT

Purpose

The TAMEL.PORT statement defines the port number on which the workstation server waits for connection requests. This statement is used with the CNMTAMEL task in the DUILPFMEM member.

Syntax

The TAMEL.PORT statement has the following syntax:

TAMEL.PORT



where:

PORT = *port*

Defines the port number. The default is 4020.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- If the CNMTAMEL task on two NetView programs are to use TCP/IP and both are using the same TCP/IP stack, specify different ports to avoid a conflict.

TAMEL.SOCKETS

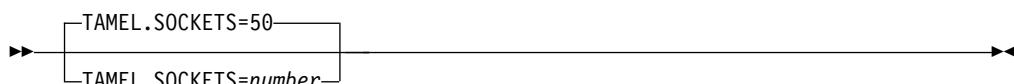
Purpose

The TAMEL.SOCKETS statement specifies the maximum number of simultaneous NETCONV sessions. This statement is used with the CNMTAMEL task in the DUILPFMEM member.

Syntax

The TAMEL.SOCKETS statement has the following syntax:

TAMEL



where:

number

Specifies the maximum number of simultaneous NETCONV sessions. The default is 50.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

TAMEL.TCPNAME

Purpose

The TAMEL.TCPNAME statement defines the TCP/IP address space name for CNMTAMEL (member DUIPPMEM).

Syntax

The TAMEL.TCPNAME statement has the following syntax:

TAMEL.TCPNAME

►►—TAMEL.TCPNAME=*name*————►►

where:

name

Defines the TCP/IP address space name.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

TAMEL.TTLS

Purpose

The TAMEL.TTLS statement specifies whether the z/OS Communications Server Application Transparent Transport Layer Security (AT-TLS) function is used to secure NETCONV TCP/IP sessions. The value that is specified also indicates if all NETCONV sessions are required to be secure.

Syntax

The TAMEL.TTLS statement has the following syntax:

TAMEL.TTLS



where:

ANY

Indicates that the z/OS Communications Server AT-TLS function is to be used to determine whether connections need to be secure. If a secure connection is needed, then a corresponding policy must be defined in the z/OS Communications Server Policy Agent. The AT-TLS function must be enabled and the z/OS Communications Server Policy Agent must be configured and active. Connections that do not have a corresponding policy are not secured.

NONE

Indicates that the z/OS Communications Server AT-TLS function is not to be used to secure any connections. This is the default value.

REQUIRE

Indicates that the z/OS Communications Server AT-TLS function is to be used to secure all connections. The AT-TLS function must be enabled and the z/OS Communications Server Policy Agent must be configured and active. Connections that do not have a corresponding policy that secures the connection are rejected and fail.

Usage notes

- If you specify TAMEL.TTLS=ANY or TAMEL.TTLS=REQUIRE, ensure that the AT-TLS function is enabled on your system. As part of enabling AT-TLS, configure the z/OS Communications Server Policy Agent (started task name is PAGENT). A sample configuration for the z/OS Communications Server Policy Agent is in the /usr/lpp/netview/v6r2m1/samples directory. See the znetview_at-tls_readme.txt file for instructions on using this sample.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

TAMEL.USETCPIP

Purpose

The TAMEL.USETCPIP statement specifies whether the CNMTAMEL task is to use TCP/IP.

Syntax

The TAMEL statement has the following syntax:

TAMEL.USETCPIP



where:

N0 Specifies that the CNMTAMEL task does not use TCP/IP.

YES

Specifies that the CNMTAMEL task is to use TCP/IP.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE TAMEL command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

TASK.taskname.INIT

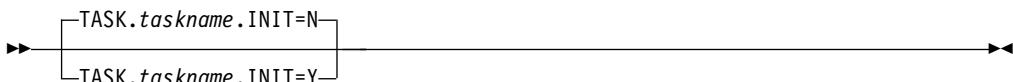
Purpose

The TASK.taskname.INIT statement specifies when a task begins.

Syntax

The TASK.taskname.INIT statement has the following format:

TASK.taskname.INIT



where:

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.

Usage notes

- The TASK statements in the CNMSTYLE and CNMSTASK members define a task to the NetView program. TASK statements can be split between multiple files: the CNMSTYLE member contains the TASK statements that are most commonly modified. Do not modify the TASK statements in the CNMSTASK member. For example, for the DSLOG task, the CNMSTYLE member contains the INIT statement and the CNMSTASK member contains the MOD, MEM, and PRI statements.
- To define optional tasks (OPTs) and data service tasks (DSTs), use the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. If you do not want to define OPT and DST tasks in a CNMSTYLE member, use the START TASK command to dynamically start an OPT or DST task. Refer to the online help for the START command for the syntax of the task operand.

Related statements

DSTINIT

TASK.*taskname*.MEM

Purpose

The TASK.*taskname*.MEM statement specifies the user-defined initialization member name that is used by the specified task. The task is responsible for the format and contents of the specified member. For NetView program tasks, the initialization member is processed only when MOD=DSIZDST. For more information, see “INIT.EMAAUTO” on page 117.

Syntax

The TASK.*taskname*.MEM statement has the following format:

TASK.*taskname*.MEM

►—TASK.*taskname*.MEM=*membername*—►

where:

membername

Indicates the user-defined initialization member name. This is an optional operand.

Usage notes

- The TASK statements in the CNMSTYLE and CNMSTASK members define a task to the NetView program. TASK statements can be split between multiple files: the CNMSTYLE member contains the TASK statements that are most commonly modified. Do not modify the TASK statements in the CNMSTASK member. For example, for task DSILog, the CNMSTYLE member contains the INIT statement and the CNMSTASK member contains the MOD, MEM, and PRI statements.
- To define optional tasks (OPTs) and data service tasks (DSTs), use the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. If you do not want to define OPT and DST tasks in a CNMSTYLE member, use the START TASK command to dynamically start an OPT or DST task. Refer to the online help for the START command for the syntax of the task operand.

Related statements

DSTINIT

TASK.*taskname*.MOD

The TASK.*taskname*.MOD statement specifies the module that runs a task.

The TASK.*taskname*.MOD statement has the following format:

TASK.*taskname*.MOD

►—TASK.*taskname*.MOD=*modulename*—►

where:

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.

Usage Notes:

- The TASK statements in the CNMSTYLE and CNMSTASK members define a task to the NetView program. TASK statements can be split between multiple files: the CNMSTYLE member contains the TASK statements that are most commonly modified. Do not modify the TASK statements in the CNMSTASK member. For example, for the DSILog task, the CNMSTYLE member contains the INIT statement and the CNMSTASK member contains the MOD, MEM, and PRI statements.
- To define optional tasks (OPTs) and DSTs, use the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. If you do not want to define OPTs and DSTs in a CNMSTYLE member, 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.

Related statements: DSTINIT

TASK.taskname.PRI

The TASK.taskname.PRI statement defines the dispatching priority of the specified task in relation to other subtasks running in this NetView program.

The TASK.taskname.PRI statement has the following format:

TASK.taskname.PRI



where:

9|n

Indicates a number that defines the dispatching priority of the specified task in relation to other subtasks that are running in this NetView program. This is an optional operand. The value of *n* can be from 1 - 9. The value of 1 is the highest priority that 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 to NetView tasks are priority 4.

Usage Notes:

- The TASK statements in the CNMSTYLE and CNMSTASK members define a task to the NetView program. TASK statements can be split among multiple files: the CNMSTYLE member contains the TASK statements that are most commonly modified. Do not modify the TASK statements in the CNMSTASK member. For example, for task DSILog, the CNMSTYLE member contains the INIT statement and the CNMSTASK member contains the MOD, MEM, and PRI statements.

- To define optional tasks (OPTs) and data service tasks (DSTs), use the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*. If you do not want to define OPTs and DSTs in a CNMSTYLE member, 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.

Related statements: DSTINIT

TCPCONN.DASD.*stackname*

The TCPCONN.DASD.*stackname* statement specifies filters for the collection of inactive TCP/IP connection data. Each DASD statement specifies whether data for a specific set of connections is kept on DASD after those connections become inactive. This connection data can then be queried with the TCPCONN QUERY command.

The TCPCONN.DASD.*stackname* statement has the following syntax:

TCPCONN.DASD.*stackname*

```
►—TCPCONN—.—DASD—.—stackname—.—suffix—=—filter————►
```

NOT

where:

stackname

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

suffix

A suffix determining the order in which the TCPCONN.DASD.*stackname* statements are processed. If a connection matches multiple TCPCONN.DASD.*stackname* statements, only the first applies.

filter

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

locaddr/locport,remaddr/remport

where:

locaddr

Specifies the local IP address of the connection.

locport

Specifies the local port of the connection.

remaddr

Specifies the remote IP address of the connection.

remport

Specifies the remote port of the connection.

You can use a single asterisk for any of these values, representing any address or port. The optional keyword NOT specifies that connections matching the filter string are not kept. (By default, inactive connections that do not match any TCPCONN.DASD.*stackname* statement are not kept.) See the CNMSTYLE member for coding examples.

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

TCPCONN.DSRBO

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

The TCPCONN.DSRBO statement has the following syntax:



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 DSRBs are available, the requests are queued. The default value is 10.

Usage Note:

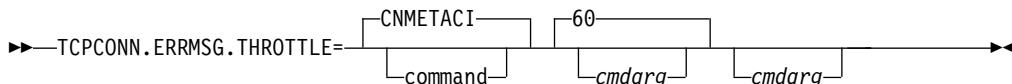
- If you change the *number*, update the STRNO keyword on the BLDVRP macro used to create the LSR pool in CNMSJM01.

TCPCONN.ERRMSG.THROTTLE

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

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

The TCPCONN.ERRMSG.THROTTLE variable has the following syntax:



where:

command

Is a command that TCPCONN is to call whenever an error is encountered writing records to the VSAM database. The default command is the sample REXX procedure CNMETACI.

cmdarg

Is a command argument for the command that you specify. You can specify

more than one *cmdarg*. If no command was specified, the default command CNMETACI is called, with a default command argument of 60.

Usage Notes:

- During TCPCONN start processing, the NetView program determines whether the *command* named in TCPCONN.ERRMSG.THROTTLE is a REXX procedure. If so, a LOADCL command is issued for the named procedure to improve performance.
- The *command* named in TCPCONN.ERRMSG.THROTTLE is not subject to authorization checking.
- The following example shows all errors being sent to an authorized receiver.
TCPCONN.ERRMSG.THROTTLE=PIPE SAFE * | ROUTE AUTHRCVR

TCPCONN.GTF.*stackname*

The TCPCONN.GTF.*stackname* statement specifies whether GTF tracing of incoming records and buffers occurs. The TCPCONN.GTF.*stackname* statement has the following syntax:

TCPCONN.GTF.*stackname*

►—TCPCONN.GTF.*stackname*=
 [NO]
 [YES]—►

where:

stackname

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

NO Do not perform GTF tracing.

YES

Perform GTF tracing.

Note: This statement applies only if collection of TCP/IP connection data is started automatically during NetView initialization using the INIT.TCPCONN statement. To control GTF tracing when starting data collection manually, use the GTF key word on the TCPCONN START command. See the *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)* for more information.

TCPCONN.HASHSIZE.*stackname*

Purpose

The TCPCONN.HASHSIZE.*stackname* statement specifies the approximate maximum number of active connections.

Syntax

The TCPCONN.HASHSIZE.*stackname* statement has the following syntax:

TCPCONN.HASHSIZE.*stackname*

►—TCPCONN.HASHSIZE.*stackname*=*connections*—►

where:

stackname

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

connections

The number of active connections.

TCPCONN.KEEP.*stackname*

Purpose

The TCPCONN.KEEP.*stackname* statement specifies filters for collection of active TCP/IP connection data. Each KEEP statement specifies whether data for a specific set of connections is kept in storage while those connections are active. This connection data can then be queried with the TCPCONN QUERY command.

Syntax

The TCPCONN.KEEP.*stackname* statement has the following syntax:

TCPCONN.KEEP.*stackname*

►►TCPCONN—.KEEP—.*stackname*—.*suffix*—filter►►

NOT

where:

stackname

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

suffix

A suffix determining the order in which the TCPCONN.KEEP.*stackname* statements are processed. If a connection matches multiple TCPCONN.KEEP.*stackname* statements, only the first applies.

filter

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

locaddr/locport,remaddr/remport

where:

locaddr

Specifies the local IP address of the connection.

locport

Specifies the local port of the connection.

remaddr

Specifies the remote IP address of the connection.

remport

Specifies the remote port of the connection.

You can use a single asterisk for any of these values, representing any address or port. The optional keyword NOT specifies that connections matching the filter

string are not kept. (By default, active connections that do not match any TCPCONN.KEEP.*stackname* statement are not kept.) See the CNMSTYLE member for coding examples.

Note: TCPCONN.KEEP.*stackname* applies only to the collection of data for active connections. Use the TCPCONN.DASD.*stackname* statement to specify which connections you want to keep on DASD after they become inactive.

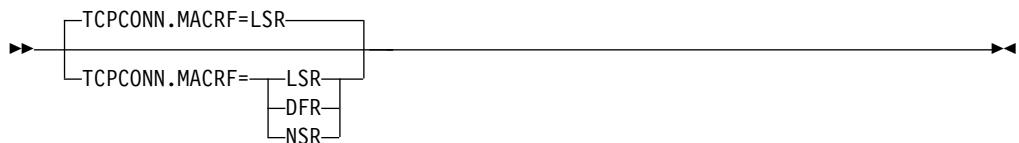
TCPCONN.MACRF

Purpose

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

Syntax

The TCPCONN.MACRF statement has the following syntax:



where:

LSR

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

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.

Note: Do not use the DFR option unless instructed by IBM Software Support.

NSR

Indicates that the data set does not use shared resources.

Note: Do not use the NSR option unless instructed by IBM Software Support.

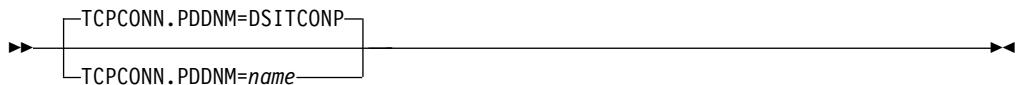
TCPCONN.PDDNM

Purpose

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

Syntax

The TCPCONN.PDDNM statement has the following syntax:



where:

name

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

Usage notes

- If necessary, specify the VSAM password in CNMSTPWD.

TCPCONN.ROWSA.*stackname*

Purpose

The TCPCONN.ROWSA.*stackname* statement specifies the maximum number of active connections for a given stack that can be stored for the NetView for z/OS Enterprise Management Agent.

Syntax

The TCPCONN.ROWSA.*stackname* statement has the following syntax:



where:

stackname

The name of the TCP/IP stack to be used.

maxnum

The maximum number of connections that can be stored for the specified stack. The valid range is 1 - 500000.

Usage notes

- For performance reasons, do not collect connections for the same stack across multiple NetView programs in the same logical partition (LPAR).
- The NACMD.ROWSCONNACT statement specifies the maximum number of active connections for all stacks that can be stored for the NetView for z/OS Enterprise Management Agent. Note the maximum value of the NACMD.ROWSCONNACT statement when setting the value for the TCPCONN.ROWSA.*stackname* statement.
- If no value is specified for the TCPCONN.ROWSA.*stackname* statement, the NetView program will set the maximum value to the default value of 200000 active connections for the specified *stackname*.

TCPCONN.ROWSD.*stackname*

Purpose

The TCPCONN.ROWSD.*stackname* statement specifies the maximum number of active DVIPA connections for a given stack that can be stored for the NetView for z/OS Enterprise Management Agent.

Syntax

The TCPCONN.ROWSD.*stackname* statement has the following syntax:

►►—TCPCONN.ROWSD.*stackname*=*maxnum*————►►

where:

stackname

The name of the TCP/IP stack to be used.

maxnum

The maximum number of DVIPA connections that can be stored for the specified stack. The valid range is 1 - 500000.

Usage notes

- The NACMD.ROWSDVCONN statement specifies the maximum number of active DVIPA connections for all stacks that can be stored for the NetView for z/OS Enterprise Management Agent. Note the maximum value of the NACMD.ROWSDVCONN statement when setting the value for the TCPCONN.ROWSD.*stackname* statement.
- If no value is specified for the TCPCONN.ROWSD.*stackname* statement, the NetView program will set the maximum value to the default value of 200000 active DVIPA connections for the specified *stackname*.

TCPCONN.SDDNM

Purpose

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

Syntax

The TCPCONN.SDDNM statement has the following syntax:

►►—

TCPCONN.SDDNM=DSITCONS
TCPCONN.SDDNM= <i>name</i>

————►►

where:

name

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

Usage notes

- If necessary, specify the VSAM password in CNMSTPWD.

TCPname

Purpose

The TCPname statement specifies the TCP name as a global variable.

Syntax

The TCPname statement has the following syntax:

TCPname

►►TCPname—=—*TCP_name*————►►

where:

TCP_name

Indicates a 1 - 5 character TCP name.

Usage notes

- If you set the system symbolic &CNMTCPPN in SYS1.PARMLIB member IEASYMxx, do not update this statement.

TCPserver

Purpose

The TCPserver statement defines the TCP/IP server to use for your commands.

Syntax

The TCPserver statement has the following syntax:

TCPserver

►► [TCPserver=UNIX] —————►►

►► [TCPserver=*server_name*] —————►►

 |————[TSO]————|

 |————[UNIX]————|

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.

TOWER

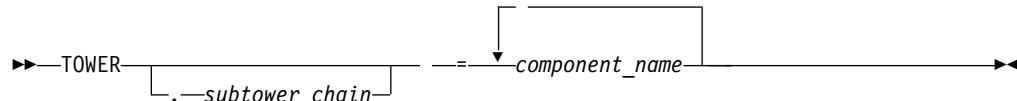
Purpose

The TOWER statement activates NetView components and other products.

Syntax

The TOWER statement has the following syntax:

TOWER



where:

subtower_chain

Specifies an optional sequence of subtower names, where each name is separated by a period. Each name can be up to 16 characters long. If the subtower_chain name sequence is not enabled, then the entire TOWER statement is ignored.

component_name

Identifies the tower name. Tower names are alphanumeric (includes @, #, and \$) and are from 1 - 16 characters.

The following towers are provided with the NetView product:

Tower Description

AMI Enables the Application Management Instrumentation.

AON Enables network automation (AON component).

Subtower

Description

SNA SNA automation (AON/SNA)

Subtower

Description

X25 AON/SNA X.25 support

TCP TCP/IP automation (AON/TCP)

Note: Do not enable the IPMGT tower if you have the AON TCP subtower enabled.

Subtower

Description

IDS Intrusion Detection Services (IDS) support

DISCOVERY

Enables the discovery of sysplexes, z/OS systems, coupling facilities, TCP/IP stacks, TCP/IP subplexes, and NetView applications.

Subtower

Description

INTERFACES

Enables the discovery of stack interface information.

Subtower

Description

OSA Enables the discovery of OSA channels and ports.

HIPERSOCKETS

Enables the discovery of HiperSockets™ configuration and status information.

TELNET

Enables the discovery of Telnet servers and ports.

DVIPA

Enables the collection of dynamic virtual IP addressing (DVIPA) definition and status data.

Subtower

Description

DVCONN

Enables the collection of DVIPA connections data.

DVRROUT

Enables the collection of VIPA routes and distributed DVIPA connection routing data.

DVTAD

Enables collection of DVIPA sysplex distributors, distributed DVIPA targets, distributed DVIPA server health statistics, and distributed DVIPA statistics (if enabled).

GRAPHICS

Enables the NetView Management console.

Subtower

Description

SNATM

SNA Topology Manager.

IPMGT

Enables IP Management

Note: Do not enable the IPMGT tower if you have the AON TCP subtower enabled.

Subtower

Description

ACTMON

Performs active monitoring for IP resources without enabling the AON component (AON tower).

IDS Enables Intrusion Detection automation without enabling the AON component (AON tower).

MSM Enables the MultiSystem Manager.

Subtower

Description

ITNM IBM Tivoli Network Manager feature.

OPN Open feature.

MVScmdMgt

Enables MVS command management.

The MVS Command Management function is deprecated and is replaced by the MVS Command Revision function. The MVS Command Revision function does not require the MVScmdMgt tower to be enabled. For more information, see the *IBM Tivoli NetView for z/OS Automation Guide*.

NLDM

Enables the session monitor.

NPDA

Enables the hardware monitor.

NVSOA

Enables the Web Services Gateway function.

SA Enables System Automation for z/OS.

TCPIPCOLLECT

Enables the collection of TCP/IP connection and packet trace data from z/OS Communications Server.

Subtower

Description

TCPCCONN

Required for the collection of TCP/IP connection data using the TCPCCONN START and TCPCCONN STOP commands.

PKTS Required for the collection of TCP/IP packet trace data using the PKTS START and PKTS STOP commands, and for the collection of OSA packet trace data.

TEMA

Enables the NetView program to communicate with the Tivoli NetView for z/OS Enterprise Management Agent.

Usage Note: Do not enable the TEMA tower unless you are installing the Tivoli NetView for z/OS Enterprise Management Agent. Only enable the TEMA tower on one NetView program for each LPAR.

Subtower

Description

CONINACT

Enables the collection and display of inactive TCP/IP connections.

CONNACT

Enables the collection and display of active TCP/IP connections.

DVCONN

Enables the display of DVIPA connections.

DVDEF

Enables the display of DVIPA definition and status data.

DVROUT

Enables the display of VIPA route data.

DVTAD

Enables the display of distributed DVIPA data.

HEALTH

Enables the collection and display of NetView task data.

HIPERSOCKETS

Enables the display of HiperSockets configuration and status information. (Requires RODM.)

OSA Enables the display of OSA channels and ports information.
(Requires RODM.)**SESSACT**

Enables the collection and display of active sessions.

The SESSACT subtower is only supported in one NetView program per system.

SYSPLEX

Enables the display of stack configuration and status data.

TELNET

Enables the display of Telnet servers and Telnet server port information.

Usage notes

- To disable a tower, either remove the name or prefix the name with an asterisk. For example, 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, see the CNMSTYLE member and the *IBM Tivoli NetView for z/OS Installation: Getting Started* manual.
- Consider the sample statements:
`TOWER.DISCOVERY = INTERFACES TELNET
TOWER.DISCOVERY.INTERFACES = OSA HIPERSOCKETS.`

If the DISCOVERY tower is enabled, then the first statement enables both the DISCOVERY.INTERFACES and DISCOVERY.TELNET subtowers. The second statement then enables the DISCOVERY.INTERFACES.OSA and DISCOVERY.INTERFACES.HIPERSOCKETS subtowers.

- Tower chains of any length can be created.
- To add or delete towers and subtowers, refer to “MODIFY.TOWER” on page 128.

TRACE.MODE**Purpose**

The TRACE.MODE statement specifies in which area data is to be logged. Specifying a value other than INT causes any tracing done at early initialization to be lost.

Syntax

The TRACE statement has the following syntax:

TRACE.MODE



where:

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.

Usage notes

- Tracing might have started very early at initialization, using the parameters MODE=INT, SIZE=trsizE, OPTION=ALL, where trsizE is the size specified on the TRSIZE procedure variable. Specifying TRACE.MODE=GTF or a different TRACE.SIZE value than that used in the TRSIZE parameter in the NetView CNMSJ009 startup procedure causes the NetView trace to be restarted and, as a result, all early trace records are lost.
- Specifying TRSIZE=0 in the NetView startup procedure disables early tracing. The trace can still be started by the TRACE statements in the CNMSTYLE member.
- By default, TRACE is on, internal, with size 4000, and options DISP, PSS, QUE, STOR, and UEXIT.
- To run without trace, and to stop any tracing started at early initialization, specify TRACE.MODE=*NONE*.
- If you notice a significant increase in processor utilization during initialization, you can change the TRACE options or start the trace after NetView initialization is complete.
- For more information, refer to the online help for the TRACE command.

TRACE.MODFILT

Purpose

The TRACE.MODFILT statement filters out unwanted modules from the trace. Only use this option when module entry/exit tracing is enabled (OPTION=ALL or OPTION=MOD).

Syntax

The TRACE.MODFILT statement has the following syntax:

TRACE.MODFILT

►►TRACE.MODFILT=*module_name*►►

where:

module_name

Specifies the module name

Usage notes

- For some NetView program problems, IBM Software Support might ask you to specify certain MODFILT values so that only modules related to the problem are traced. Eliminating the tracing of modules not related to the problem can greatly reduce the likelihood of the trace wrapping.

TRACE.MONOPER

Purpose

The TRACE.MONOPER statement queues messages to the monitoring operator task.

Note: Only use the TRACE.MONOPER statement when IBM Software Support requests you to use it. If used incorrectly, this statement can cause the NetView program to run out of storage and end because of too many messages being queued to the monitoring operator task. Therefore, use the TRACE.MONOPER statement with extreme caution. It is a debugging aid, and even when used correctly, its use can degrade performance.

Syntax

The TRACE.MONOPER statement has the following syntax:

TRACE.MONOPER

►► [TRACE.MONOPER=NONE] [TRACE.MONOPER=*operator_task*] ►►

where:

operator_task

Specifies a monitoring operator task.

Usage notes

- When the monitoring operator task other than the default of NONE is specified (for example, when an *operator_task* value of OPER2 is specified), the NetView program sends messages to the specified operator task that display in real time the trace entries that are currently being traced by the TRACE command. The messages are written to the NetView log and displayed at the operator task. These trace entries assist IBM Software Support when debugging problems. Tracing is disabled for the *operator_task*, and the *operator_task* cannot be present in the list of task values specified for the TASK keyword.
- For more information, refer to the online help for the TRACE command.

TRACE.OPTION

Purpose

The TRACE.OPTION statement specifies which options are traced. Each option identifies an internal event type that is traced.

Syntax

The TRACE statement has the following syntax:

TRACE.OPTION



where:

DISP

Indicates the dispatching of tasks, which includes 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 intertask queueing of buffers using the DSIMQS macro.

STOR

Indicates the getting and freeing of storage.

TCP

Indicates IP services related calls.

UEXIT

Indicates installation exit calls:

- DSIEX01 through DSIEX19
- 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)

Usage notes

- To aid in debugging NetView initialization problems, the NetView program starts tracing early in the initialization process using the parameters MODE=INT, SIZE=*trsizE*, OPTION=AL, where *trsizE* is the size specified on the TRSIZE procedure variable. Specifying different trace options using the TRACE.OPTION statement can stop early tracing or override trace options that were in effect for early initialization tracing. However, it does not prevent the early tracing from starting.
- Specifying TRSIZE=0 in the NetView startup procedure disables early tracing. The trace can still be started by the TRACE statements in the CNMSTYLE member.
- By default, TRACE is on, internal, with size 4000, and options DISP, PSS, QUE, STOR, and UEXIT.
- If you notice a significant increase in processor utilization during initialization, you can change the TRACE options or start the trace after NetView initialization is complete.
- For more information, refer to the online help for the TRACE command.

TRACE.SAFopt

Purpose

The TRACE.SAFopt statement specifies the types of SAF calls to trace.

Syntax

The TRACE.SAFopt statement has the following syntax:

TRACE.SAFopt

►►—TRACE.SAFopt=*options*—————►►

where:

options

Can be one of the following values:

- ALL
- AUTH
- EXTRACT
- FASTAUTH
- LIST
- STAT
- TOKENMAP
- TOKENXTR
- VERIFY

Usage notes

- For more information, refer to the online help for the TRACE command.

TRACE.SAFrc

Purpose

The TRACE.SAFrc statement traces calls made to an SAF product.

Syntax

The TRACE.SAFrc statement has the following syntax:

TRACE.SAFrc

►►TRACE.SAFrc=*calls*————►►

where:

calls

Can be one of the following values:

- ALL
- FAIL
- NONE

Usage notes

- For more information, refer to the online help for the TRACE command.

TRACE.SIZE

Purpose

The TRACE.SIZE statement specifies the number of pages to use to define the trace data space for the MODE=INT option.

Syntax

The TRACE.SIZE statement has the following syntax:

TRACE.SIZE

►► [TRACE.SIZE= *]
[TRACE.SIZE=*pages*]————►►

where:

pages

Indicates the number of pages. The maximum size is 524286.

- * Specifies to use the size of the trace table defined during early initialization. If tracing was not started at early initialization, TRACE.SIZE= * results in the default value of 4000 (4 KB page size).

Usage notes

- If SIZE is specified with MODE=GTF, SIZE is ignored.

- Specifying a value different than the size used to define the trace table at early initialization causes any tracing done at early initialization to be lost.
- For more information, refer to the online help for the TRACE command.

TRACE.TASK

Purpose

The TRACE.TASK statement specifies a task name or a task type to be traced.

Syntax

The TRACE.TASK statement has the following syntax:

TRACE.TASK



```
►►TRACE.TASK=task_type►►
```

where:

task_type

Specifies a task name or a task type. The following values for *task_type* are valid:

- ALL
- HCT
- MNT
- NNT
- OPT
- OST
- PPT
- VOST

Usage notes

- For more information, refer to the online help for the TRACE command.

transMember

Purpose

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 TRANSMMSG command. Stop and restart the NetView program to implement the changes.

Syntax

The transMember statement has the following syntax:

transMember

►—transMember—=—*membername*—►

where:

membername

The DSIMSG member that contains the message translations.

transTbl

Purpose

The transTbl statement defines a character translation set to the NetView program. Stop and restart the NetView program to implement the changes.

Syntax

The transTbl statement has the following syntax:

transTbl

►—
 |—transTbl=DSIEBCDC—
 |—transTbl=DSIKANJI—
►

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.
- Ensure that the transTbl value for the log printer program is the same value as that 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.

VTAMCP.USE

Purpose

The VTAMCP.USE statement specifies whether the NetView SNA MS transport running under a specific NetView program can receive MDS-MUs with the VTAM control point (CP) name as the destination. Use the VTAMCP.USE statement when you have multiple NetView programs running under a single VTAM program.

Syntax

Changes to the VTAMCP.USE statement do not take effect until you stop and restart the NetView program.

VTAMCP.USE



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 *IBM 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.
- Use the same value for VTAMCP.USE as you specified for the CNMI statement, unless instructed otherwise.

WBSORDER

Purpose

The WBSORDER statement defines the default order in which URLs and titles for web sites that customers want to access using the NetView WEBSITES command are to be displayed.

Syntax

The WBSORDER statement has the following syntax:

WBSORDER



where:

descriptor.suffix

The descriptor and suffix defined in the WEBSITE statement.

Usage notes

- To implement default order changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE WEBSITE command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- The default order specified here can be overridden in the initial command processing of the operator by defining the override values in a KEEP name WBSORDER. This example specifies that four URLs are to be displayed for this operator:


```

WBSORDER = NETVIEW.A NETVIEW.B NETVIEW.C netview.z
Do while WBSORDER ~= ''
  Parse var WBSORDER thisentry '' WBSORDER
  'PIPE var thisentry',
  '| safe WBSsafe append'
End
'pipe (end;) safe WBSsafe',
'| keep WBSORDER * nospill'
      
```
- To verify that the default order is specified correctly, issue BROWSE WBSORDER from a NetView operator console.
- To verify the contents of the KEEP defined for a particular operator, issue PIPE KEEP WBSORDER | CONS from the console for that operator.

WEB

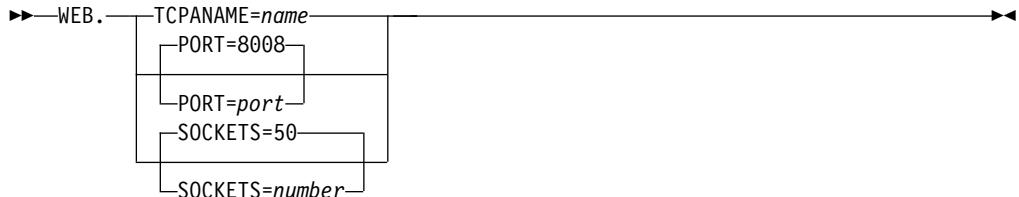
Purpose

The WEB statement defines TCP/IP definitions for NetView web access.

Syntax

The WEB statement has the following syntax:

WEB



where:

TCPNAME = 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 web browser sessions that can connect concurrently. This is used by the web access interface task (DSIWBTSK). The default is 50.

Usage notes

To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE WEB command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

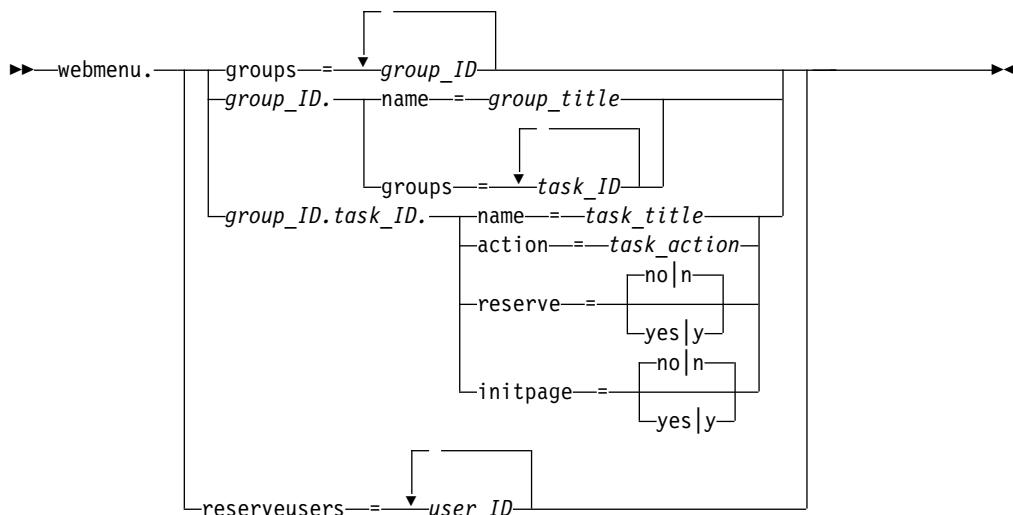
webmenu

Purpose

The webmenu statement defines settings for the NetView web application. This statement is in the included member CNMSTWBM.

Syntax

The webmenu statement has the following syntax:

webmenu

where:

groups

Specifies the groups of tasks on the portfolio, one ID for each task group, separated by spaces. The order of the group IDs specified on this keyword determines the order in which the task groups are listed in the portfolio.

group_ID.

Defines a group of tasks.

group_ID.name

Specifies the title that is displayed in the portfolio for a task group.

group_ID.groups

Specifies the tasks in a task group, one ID for each task, separated by spaces. The order of the task IDs specified on this keyword determines the order of the tasks within the task group in the portfolio.

group_ID.task_ID

Defines a task within a task group.

group_ID.task_ID.name

Specifies the title that is displayed in the portfolio for the *group_ID.task_ID*. task. This keyword is required.

group_ID.task_ID.action

Specifies the action that is used by the web application to run the *group_ID.task_ID*. task. This keyword is required. For NetView product values, do not change this keyword. You can add your own entries to call external web addresses.

When the protocol of the web address of your Launch Sample URL action is http or https, specify the following action:

`http://www.YourCompany.com`

When the protocol of the web address of your Launch Sample URL action is mailto, specify the following action:

`mailto:user@yourcompany.com`

When the protocol of the web address of your Launch Sample URL action is FTP, specify the following action:

`ftp://www.YourCompany.com`

group_ID.task_ID.reserve

Specifies whether the *group_ID.task_ID*. task is a reserved task. This keyword is optional. Valid values are yes, y, no, and n. If the value is yes or y, the task is a reserved task. A reserved task is displayed in the portfolio only for the user IDs specified by the **reserveusers** keyword. If this keyword is not present for a task, is not set, or has a value of no or n, the task is not a reserved task.

group_ID.task_ID.initpage

Specifies whether the *group_ID.task_ID*. task is the first task displayed in the web application after a user signs in to the web application. This keyword is optional. Valid values are yes, y, no, and n. If this keyword is set to yes or y for a task, that task is displayed first when a user signs in to the web application. If this keyword is set to yes or y for more than one task, the first task with a value of yes or y is displayed when a user signs in. If this keyword is not set to yes or y for any task, the Welcome view is displayed when a user signs in.

reserveusers

Specifies one or more users that are authorized to access the reserved tasks in the portfolio, one ID for each user, separated by spaces, commas, or semicolons. This keyword is optional. If this keyword has a null value or is not present, no users are authorized to access reserved tasks. For users that are not authorized to access reserved tasks, the tasks are not displayed in the portfolio.

Usage notes

The webmenu definition statement can extend to column 80 and, if necessary, be continued to subsequent lines by leaving the first column of each subsequent line blank. You must begin typing the continued statement in column 2 of each continuation line.

WEBSITE

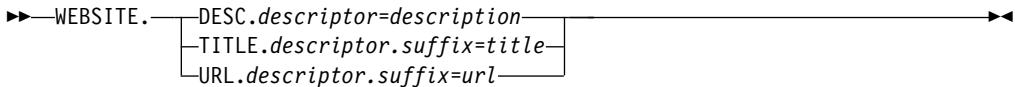
Purpose

The WEBSITE statements define the URLs and titles for web sites that customers can access using the NetView WEBSITES command.

Syntax

The WEBSITE statement has the following syntax:

WEBSITE



where:

DESC.descriptor=description

Defines a descriptor to be used in subsequent WEBSITE statements. This descriptor provides a grouping mechanism for the WEBSITES command. The *descriptor* is 1 - 8 non-blank alphanumeric or national (@, #, or \$) characters. The *description* is freeform text that is to be displayed when WEBSITES TYPE=? is issued. Operators can issue WEBSITES TYPE=*descriptor* statements to see only the URLs that contain that descriptor.

TITLE.descriptor.suffix=title

Defines a title to be displayed when the WEBSITES command is issued. This title is a description of the corresponding URL. The *descriptor.suffix* connects the title to the corresponding URL when the WEBSITES command is issued. The *suffix* is 1 - 3 non-blank alphanumeric or national (@, #, or \$) characters.

URL.descriptor.suffix=url

Defines the URL to be displayed when the WEBSITES command is issued. The *descriptor.suffix* connects the URL to the corresponding title when the WEBSITES command is issued. The *suffix* is 1 - 3 non-blank alphanumeric or national (@, #, or \$) characters.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE WEBSITE command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- The *suffix* parameter is used to determine the order in which the titles and URLs are displayed within the descriptor when the WEBSITES command is issued and WBSORDER is not set. The descriptor and suffix must match for the TITLE and URL entries to be displayed correctly.

When the WEBSITES command is issued and WBSORDER is not set, the sort order is by descriptor and then suffix.

WINDOW.EXEC

Purpose

The WINDOW.EXEC statement specifies which REXX procedure is used when the EXEC action is issued from the WINDOW display panel.

Syntax

The WINDOW.EXEC statement has the following syntax:

WINDOW.EXEC

```
►►WINDOW.EXEC—=—CNMEXEC  
                  |  
                  REXX_procedure—►►
```

where:

REXX_procedure

Specifies the procedure to use. The default is the CNMEXEC sample that is supplied with the NetView program.

WLM.SubSystemName

Purpose

The WLM.SubSystemName statement activates NetView MVS workload management (WLM) support.

Syntax

The WLM.SubSystemName statement has the following syntax:

WLM.SubSystemName

```
►►WLM.—SubSystemName—=—name—►►
```

where:

name

Specifies the SubSystemName value. This name corresponds to the system instance name specified in the WLM service classification rules.

Usage notes

- For more information, refer to CNMSTYLE member.

XCF.GROUPNUM

Purpose

The XCF.GROUPNUM statement defines the 2-character suffix that defines the DSIPLEX n group that the NetView program can join. The suffix allows NetViews

within a sysplex to be divided and grouped together.

Syntax

The XCF.GROUPNUM statement has the following syntax:

XCF.GROUPNUM



where:

group_number

Specifies the 2-character suffix to create the name of the NetView group. Numeric or alphabetic characters can be used for the *group_number*. The default group number is 01. The characters FF are reserved for the NetView program (internal use only).

Usage notes

- There can be multiple NetView groups in the sysplex, but a given NetView program can belong to only one DSIPLEX mn group.
- Each group has its own master NetView program and its members do not receive notification of XCF events that are affecting members in other groups.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Examples

In this example, the PLEX1 sysplex contains NetView domains CNM01, CNM02, CNM03 and CNM04. Domains CNM01 and CNM03 have the following statement:
XCF.GROUPNUM=01

Domains CNM02 and CNM04 have the following statement:

XCF.GROUPNUM=02

This results in two DSIPLEX mn XCF groups being created in the sysplex; DSIPLEX01 with members CNM01 and CNM03, and DSIPLEX02 with members CNM02 and CNM04.

XCF.MASTDVIPA

Purpose

The XCF.MASTDVIPA statement specifies a DVIPA address that is associated with the master NetView program. This address is dynamically defined on the TCP/IP stack for the master NetView program and deleted off the stack for a former master NetView program.

Syntax

The XCFMASTDVIPA statement has the following syntax:

XCF.MASTDVIPA



where:

dvipa

Specifies the DVIPA that is associated with the master NetView program. The default value is *NONE*.

This address must fall within the range defined by VIPARANGE statements in the TCP/IP profile. For more information on VIPARANGE statements, see the *Communications Server IP Configuration Guide* and the *Communications Server IP Configuration Reference*.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.PROCSTR nn

Purpose

The XCF.PROCSTR nn statement specifies the procedures to start when taking over as the master NetView program.

Syntax

The XCF.PROCSTR nn statement has the following syntax:

XCF.PROCSTR nn

where:

jobname

Specifies the procedure name (*jobname*). The specified procedure name is started if it is not already active.

parameters

Specifies the parameters that are associated with the specified *jobname*.

Usage notes

- Use one XCF.PROCSTR nn statement for each job that runs on the master NetView program.
- The NetView program starts the procedures in the order specified by the nn variable on the XCF.PROCSTR nn statement.

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Examples

This is an example statement for RODM that generates the start command:

```
XCF.PROCSTR01 = 'RODM TYPE=C,INIT=EKGLILM'
```

XCF.RANK

Purpose

The XCF.RANK statement determines the rank of the NetView program within the sysplex, or a NetView program running as a MONOPLEX or XCFLOCAL system.

Syntax

The XCF.RANK statement has the following syntax:

XCF.RANK



where:

rank_number

Specifies the rank of the NetView program within the sysplex. The value ranges from -1 (cannot participate in the NetView group) - 250 (master NetView program). A value of 0 indicates the NetView program participates in the group, but cannot assume the master role. A value 1 - 249, with 249 being the highest, means the NetView program is master-capable. The default value is 1.

Usage notes

- Specify only one NetView program within the sysplex as the master.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.
- You can use the following command to dynamically change the rank of a NetView program:

```
PLEXCTL RANK=rankvalue
```

XCF.TAKEOVER.CLIST

Purpose

The XCF.TAKEOVER.CLIST statement specifies a user-written command to run when NetView takes over as master. This can be used to customize processing at

your installation. For example, you can issue GETTOPO or NACMD commands, depending on your configuration.

Syntax

The XCF.TAKEOVER.CLIST statement has the following syntax:

XCF.TAKEOVER.CLIST



where:

command

Specifies the name of the command to run when NetView takes over as master. The default value is *NONE*.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.CONVIP*nn*

Purpose

The XCF.TAKEOVER.CONVIP*nn* statement specifies one or more NetView management console servers to whom NetView should attempt to establish TCP/IP NETCONV connections when taking over as master. These NETCONV connections are not established until NetView assumes the master role.

Syntax

The XCF.TAKEOVER.CONVIP*nn* statement has the following syntax:

XCF.TAKEOVER.CONVIP*nn*



where:

nn Specifies a number that is used to differentiate this statement from other CONVIP*nn* statements.

ip_address

Specifies the IP address of the NetView management console server.

port_number

Specifies the port number of the NetView management console server. If *port_number* is not specified, the default value is the value specified with the TAMEL.PORT statement.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.CONVSNA*nn*

Purpose

The XCF.TAKEOVER.CONVSNA*nn* statement specifies one or more NetView management console servers that the NetView program can use to establish SNA NETCONV connections when taking over as master. These NETCONV connections are not established until the NetView program assumes the master role.

Syntax

The XCF.TAKEOVER.CONVSNA*nn* statement has the following syntax:

XCF.TAKEOVER.CONVSNA*nn*

►►XCF.TAKEOVER.CONVSNA*nn*=*lu_name*►►

where:

nn Specifies a number that is used to differentiate this statement from other CONVSNA*nn* statements.

lu_name
Specifies the name of the logical unit.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.DELAY

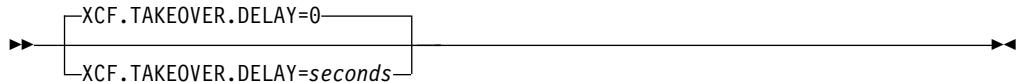
Purpose

The XCF.TAKEOVER.DELAY statement determines how long to wait before attempting data discovery when a NetView program takes over as the master NetView program.

Syntax

The XCF.TAKEOVER.DELAY statement has the following syntax:

XCF.TAKEOVER.DELAY



where:

seconds

Specifies the time to wait before attempting data discovery. The value is in the range 0 - 3600. The default value is 0.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.DURATION

Purpose

The XCF.TAKEOVER.DURATION statement determines how long a master NetView program prevents takeovers from higher-ranked NetView programs that are initializing when it assumes the master role.

Syntax

The XCF.TAKEOVER.DURATION statement has the following syntax:

XCF.TAKEOVER.DURATION



where:

minutes

Specifies the time that a master NetView program waits before it allows another NetView program to take over as master. The value is in the range 1 - 60. The default value is 5.

Usage notes

- You can use the PLEXCTL RANK=250 or PLEXCTL RANK=MASTER command from another NetView program to override the XCF.TAKEOVER.DURATION value and assume the master role. The DURATION value stops a NetView program with a defined rank of 250 from becoming master if it initializes during the DURATION period.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as

the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.INITWAIT

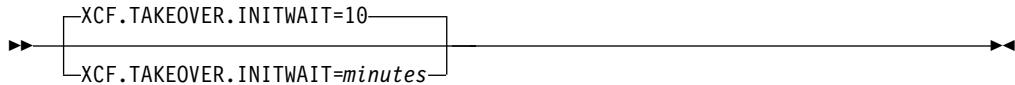
Purpose

The XCF.TAKEOVER.INITWAIT statement determines how long a backup NetView program waits to take over when it cannot detect a master NetView program in the sysplex at initialization.

Syntax

The XCF.TAKEOVER.INITWAIT statement has the following syntax:

XCF.TAKEOVER.INITWAIT



where:

minutes

Specifies the time that a backup NetView program waits before taking over as the master NetView program when it first joins an XCF group that does not have a current master. This value also applies when the rank of a backup is changed with the PLEXCTL command (specifying a rank in the range of 1 - 249) and no master is active in the group. The value is in the range 1 - 60. The default value is 10.

Usage notes

- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

XCF.TAKEOVER.NETCONVS

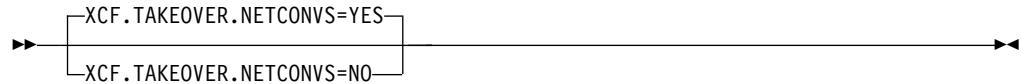
Purpose

The XCF.TAKEOVER.NETCONVS statement defines whether a NetView program taking over as master is to attempt to establish NETCONV connections and sessions that were active for a previous master.

Syntax

The XCF.TAKEOVER.NETCONVS statement has the following syntax:

XCF.TAKEOVER.NETCONVS



where:

NO Specifies not to establish the NETCONV connections and sessions.

YES

Specifies to establish the NETCONV connections and sessions. This is the default.

Usage notes

- If the previous master NetView program left the XCF group with the STOP XCFGROUP command, the new master NetView program cannot detect that it should end existing NetView connections at the previous master NetView program. The NetConv connections must be manually terminated and started at the new master NetView program. This restriction does not apply if the previous master NetView program is ending.
- To implement definition changes while the NetView program is active, change the value in the CNMSTUSR or CxxSTGEN member, and enter the RESTYLE XCF command. However, if the NetView program is already the master, changes will not take effect until the next time that this NetView program takes over as the master. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

Chapter 3. NetView Definition Statements

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 can 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. You can use this information to help determine requirements for user coding and NetView program definitions.

Use the definition statements to accomplish the procedures described in the *IBM 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)

Purpose

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.

Syntax

The A statement has the following syntax:

```
►►A— class [blank] [N] [blank] [N] [blank] [N] —comments— ►►  
      |   |   |   |  
      color  notify  high  alarm  console
```

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 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 is 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

Usage notes

- Code the alert statement in DSICNM. DSICNM changes do not take effect until you stop and restart the *nv_id*VMT task (where *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.

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

- Code 1 A statement for each message indicator class you define. You can code 4 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 *IBM 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.

ACCESS

Purpose

Use the ACCESS statement 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 host name and an optional user name on a single line, with one exception. The special character "+" can be used on a line by itself to signal universal access except for users or hosts prohibited by another ACCESS statement.

Syntax

The ACCESS statement has the following syntax:

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. For example, if using dotted notation: 127.44.44.44).

user

Specifies the username on the remote system.

Usage Notes:

- After a host is specified in an ACCESS statement, subsequent access statements must agree with the security level set for that host. For example, if "-host1" is specified, "host1" or "+host1" is not valid on a subsequent 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.

ACTION

Purpose

Use the ACTION statement to change the color of action messages that are displayed on the command facility panel. The sample member containing the ACTION statement is CNMSCNFT. You can code the ACTION statement only once.

Syntax

The ACTION statement has the following syntax:

ACTION

►►—ACTION— [*colorf*] [*attribute*] [ON *colorb*] ►►

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 program supplies the following defaults for the ACTION statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity

- Alarm on
- The following color operands can be specified:

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 following highlight operands can be specified:

BLINK

The characters blink.

REV The characters are changed to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following intensity operands can be specified:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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, TITLEOPID,
TITLESTAT, TITLETIME

ALIASMEM

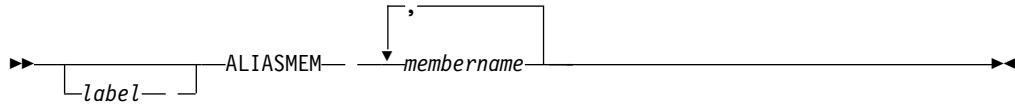
Purpose

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. A sample table of alias name translations is shipped with the NetView program as DSIPARM member DSIALTAB.

Syntax

The ALIASMEM statement has the following syntax:

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.

Usage notes

- Code this statement after the DSTINIT statement that defines DSIAINIT as the DST initialization exit.
- 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.

Related statements

COS, DSTINIT, LU, MODE, ORIGNET, TASK

AMELINIT

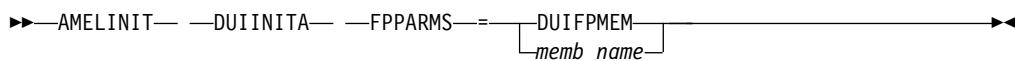
Purpose

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.

Syntax

The AMELINIT statement has the following syntax:

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.

Usage notes

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.

AUTH

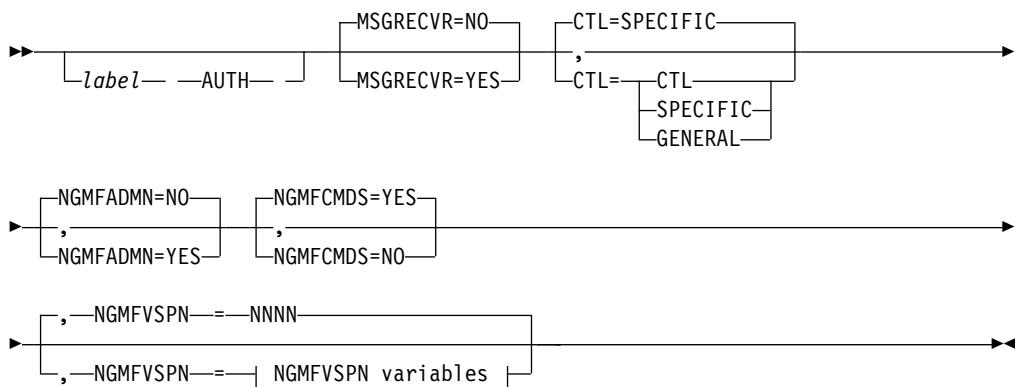
Purpose

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 the authority of an operator 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 373 and "PROFILEN" on page 383 for information on how a PROFILEN statement is associated with an operator. A sample member supplied with the NetView program is DSIPROFA.

Syntax

The AUTH statement has the following syntax:

AUTH



NGMFVSPN variables:

—span level—visible objects—restrict view info—restrict list info—

where:

Label

et 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.

Note:

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 authority of the operator to control resources. This operand must indicate one of the following values:

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 profile for the operator, or if OPSPAN=SAF, 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 profile for the operator, 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 the CNMSTYLE member 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 following NetView management console functions are controlled by this operand:

- 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. The following NetView management console commands are allowed by this operand:

- 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 authority of the operator to display NetView management console views and resources within views. The NGMFVSPN attribute specifies whether each resource, each view name, or both are to 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 are to 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 a coding error exists 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. For example, an error such as NGMFVSPN=VZNN, results 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 that 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 that 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 that 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 that 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 span of control for an operator 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 the span of control for an operator are not displayed in the views for the operator. N is the default.
- Y** Visible. Any resources not in the span of control for an operator are displayed in the views for the operator as null nodes and links.

restrict_view_info

Specifies whether an indication is given to the operator when objects not in the span of control for the operator 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 span of control for the operator.

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 span of control for the operator. 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 display because they are not in the span of control for the operator.

restrict_list_info

Specifies whether an indication is given to the operator when view names or resource names that are not in the span of control for the operator 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 span of control for the operator.

Note:

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 can use and the NGMFVSPN setting for each level of span checking.

Table 10. Examples of Span Checking Options and NGMFVSPN Settings

Desired Span Checking Level	NGMFVSPN Setting	Results
No span checking	NNNN	<p>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. Use this option for all operators who have CTL=GLOBAL in their operator profiles.</p> <p>If you do not want to use the NGMFVSPN attribute now, but think you use it later, code NGMFVSPN=NNNN as a placeholder.</p>
Span checking for views only	VNNN	<p>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.</p>
Span checking for resources only	RYNN	<p>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.</p>

Table 10. Examples of Span Checking Options and NGMFVSPN Settings (continued)

Desired Span Checking Level	NGMFVSPN Setting	Results
Span checking for all views and resources	ANYY	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.

Related statements

DOMAINS, ISPAN, OPERATOR, SECOPTS, PROFILE, PROFILEN, SPAN

C (Command List)

Purpose

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

The C statement has the following syntax:

►—C— —cmdlistname— —comments—►

where:

C Must be in column 1.

cmdlistname

Is the name of the command list that 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.

Usage notes

- 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.

CMDCLASS

Purpose

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

Related statements

KEYCLASS, OPCLASS, VALCLASS

CMDDEF

Purpose

The CMDDEF statement defines commands and command lists to the NetView program.

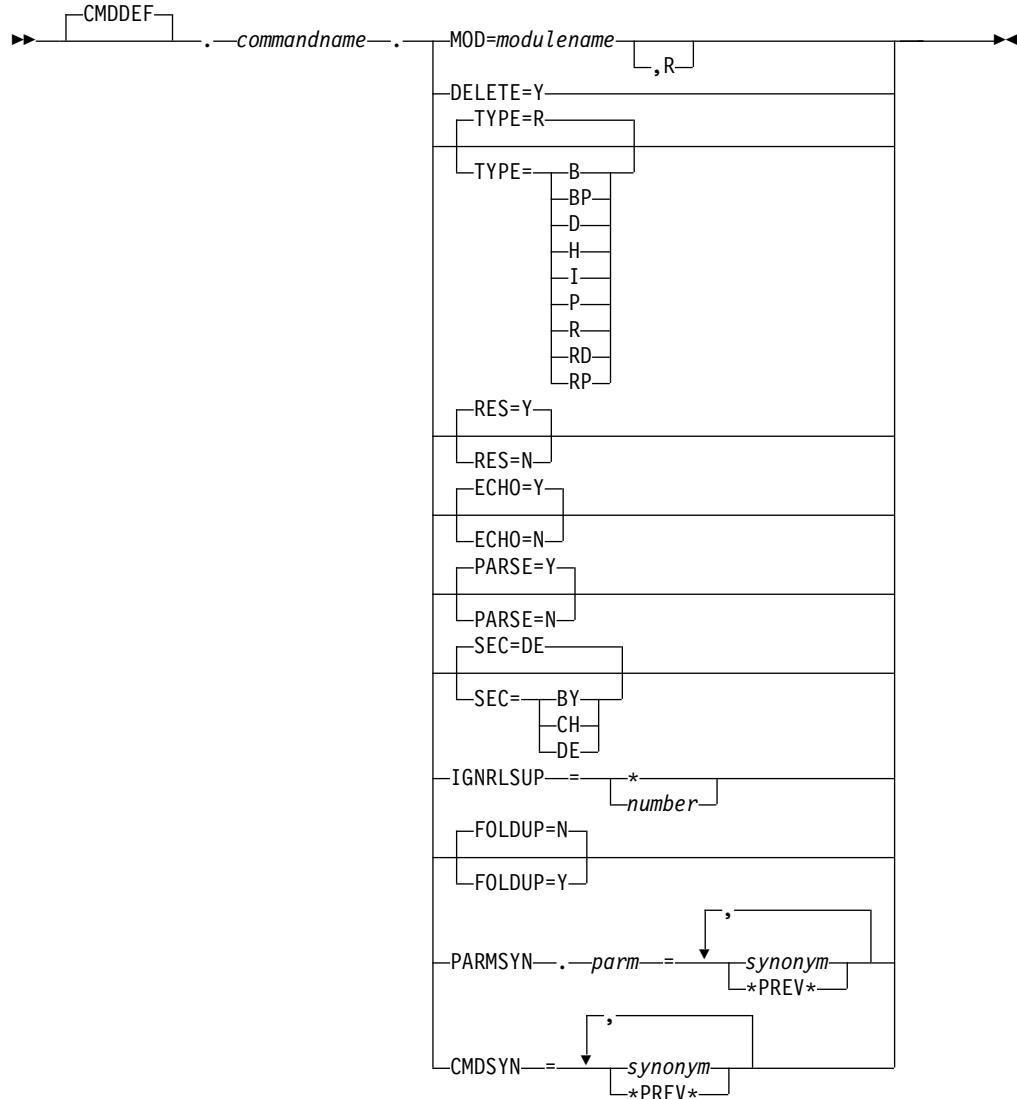
For commands, the CMDDEF statement defines the following information:

- Command name (verb)
- Module that processes the command
- Command type
- When the command module is loaded
- Whether authority verification is performed

Syntax

The CMDDEF statement has the following syntax:

CMDDEF



Operand	Synonym
MOD	MODULE
RES	RESIDENT
SEC	SECURITY

where:

CMDDEF

Identifies this statement as a command definition statement. This identifier is optional; if present, CMDDEF must begin in column 1. The period following CMDDEF is required. If the identifier CMDDEF is omitted, the period must begin in column 1.

commandname

Is the name of the command or command list. Do not use the following characters in the *commandname*:

- Blank space
- Colon (:)
- Comma (,)

- Equals (=)
- Percent (%)
- Slash (/)

All lowercase alphabetic characters are translated to uppercase. To avoid conflicts in naming, do not begin *commandname* with any of the 3-character prefixes that the NetView program uses. These prefixes include AAU, BNH, BNJ, BNK, CNM, DSI, DUI, EZL, FLC, FKX, and FKV.

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 run that command instead of the user-written command procedure.

R Indicates that this is a replacement and not a duplicate of a previously defined command in CNMCMD. Specifying the R effectively suppresses unwanted DUPLICATE COMMAND messages for those cases where command modules are intentionally being replaced in a command definition.

Note: The word REPLACE can be spelled out for documentation purposes, but no checking for spelling or syntax is done beyond the initial ,R.

DELETE=Y

Deletes a command already defined by a CMDDEF statement. Use this specification to exclude a command definition when the NetView program is started.

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, the NetView program 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 point at which the second command begins processing is dependent on the value of the CMD parameter specified on the DEFAULTS or OVERRIDE command. 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 command definitions with duplicate module names that have conflicting residency attributes, RES=Y is assumed. If the command definition is TYPE=I or TYPE=B and RES=N, 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, DSICYCLF, DSICYCTRM, DSICYOCRM, DSIFYOFP, DSIFYOLRP, DSIFYOMRP, DSIFYOSRP, DSIFYOTP, DSIFYPIF, 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 REXX command lists, the setting for TRACE determines the echo status. 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 that an entered command is not echoed.
- Y** Means that 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 calls.

- 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 about how to write command procedures, refer to *IBM Tivoli NetView for z/OS Programming: Assembler* and to *IBM Tivoli NetView for z/OS Programming: 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 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 is not to 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 is always to perform authority verification when this command is issued. This overrides AUTOSEC=BYPASS as specified on the DEFAULTS command.

Use SEC=CH to restrict access to commands that if misused can 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 is to 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 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 are valid.

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 are 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.

Note:

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 CMDDEF statement takes precedence over any value coded on the DEFAULTS command. If IGNRLSUP is not coded on the DEFAULTS or ADDCMD commands or the CMDDEF 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.

FOLDUP=N|Y

When the OVERRIDE NETVASIC=YES command is in effect, this indicates whether the command string is to be folded to uppercase before being passed to the command processor.

- N** Indicates that the command is not converted to uppercase. This is the default.
- Y** Indicates to convert the command to uppercase.

Note:

1. You can override FOLDUP=Y by prefixing *commandname* with NETVASIC when issued.
2. FOLDUP=Y takes precedence over the OVERRIDE NETVASIC=YES specification for the task that is running the command.
3. The FOLDUP function is for commands that are entered from a command line. It has no effect on commands that are called in a procedure (for example command list, REXX, PL/1, or C).

PARMSYN.*parm*=*synonym*|*PREV*

parm is the original command parameter name. *synonym* is a new 1 - 8 character parameter name. After the NetView program is recycled, it can accept either the old or the new name.

Usage Notes:

- You can specify multiple synonyms, separated by commas.
- Use *PREV* to append the synonyms defined on this PARMSYN.*parm* statement with previous PARMSYN statements for this command and this parameter. If you do not specify *PREV*, any previously defined synonyms for this command and this parameter are replaced.
- Do not use PARMSYN for VTAM, MVS, and terminal access facility (TAF) LU1 commands.
- When you use help, only the original parameter is used in the help information.
- See the CNMCMD sample for the NetView defined synonyms and then customize CNMCMU.

CMDSYN=*synonym*|*PREV*

Indicates the new name for the command or command list. The operator can use either the original command name or the new synonym.

Usage Notes:

- You can specify multiple synonyms, separated by commas.

- Use *PREV* to append the synonyms defined on this CMDSYN statement with previous CMDSYN statements for this command. If you do not specify *PREV*, any previously defined synonyms for this command are replaced.
- 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 *IBM Tivoli NetView for z/OS Customization Guide* for more information about writing help panels.
- The system console operator must always use the original command name.

See the CNMCMD sample for the NetView defined synonyms and then customize CNMCMU.

Usage notes

- The CMDDEF statement is used in CNMCMD. Stop and restart the NetView program to implement the changes.
- CMDDEF statements are processed sequentially. The last statement with the same value to the left of the equal sign is the statement that is used (except for *PREV* values). All CNMCMD members are merged before statement selection is performed.
- Use a CMDDEF statement for command lists to:
 - Define synonyms for the command list or parameters
 - Explicitly check or bypass authority checking for a particular command or command list

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

- 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 CMDDEF statement in CNMCMD for non-NetView commands.
- To improve initialization performance, do not specify default values, for example RES=Y or TYPE=R.
- You can continue definition statements that specify a keyword whose value is a comma-delimited list. In this case, the value can be continued from one line to another by ending the line to be continued with a comma, and resuming the list value in column 2 of the following line, as shown in the following example:

```
CMDDEF.FOCALPT.PARMSYN.ADDBKUP=ADBU,ADD,
ABK,AB
```

Related statements

SECOPTS

CMDLINE

Purpose

Use the CMDLINE statement 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.

Syntax

The CMDLINE statement has the following syntax:

CMDLINE

►►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 program supplies the following defaults for the CMDLINE statement:
 - Green foreground
 - Black background
 - Normal highlighting
 - Low intensity
 - Alarm off
- The following color operands can be specified:

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 following highlight operands can be specified:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following intensity operands can be specified:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

BEEP An audible alarm is sounded.

Related statements

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

CMDMDL

Purpose

The CMDMDL statement is obsolete. It is used for migration purposes only. It is replaced by the CMDDEF statement in DSIPARM member CNMCMRD. For more information, refer to the *IBM Tivoli NetView for z/OS Installation: Migration Guide*.

Related statements

CMDSYN, PARMSYN

CMDRCVR

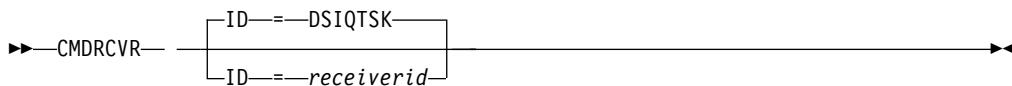
Purpose

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

Syntax

The CMDRCVR statement has the following syntax:

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 processing.

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

Purpose

The CMDSYN statement is obsolete. It is used for migration purposes only. It is replaced by the CMDDEF statement in DSIPARM member CNMCMD. For more information, refer to the *IBM Tivoli NetView for z/OS Installation: Migration Guide*.

Related statements

CMDMDL, PARMSYN

COLUMNHEAD

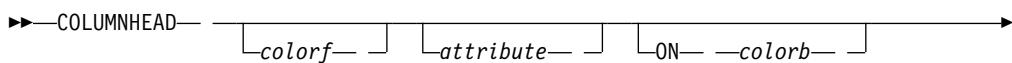
Purpose

Use the COLUMNHEAD statement to change the color of the column heading line that is displayed 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.

Syntax

The COLUMNHEAD statement has the following syntax:

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 are not 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 following color operands can be specified:

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 following highlight operands can be specified:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following intensity operands can be specified:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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

COMNTESC

Purpose

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

Related statements

CMDMDL, CMDSYN, PARMSYN

COS

Purpose

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.

Syntax

The COS statement has the following syntax:

COS

►— [label] —COS— —targname—,—targnet—,—origname————►

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

DBFULL

Purpose

Use the DBFULL statement 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. After message BNJ183I is issued, no more messages are issued until one of the following situations occurs:

- You stop the BNJDSERV task.
- You issue the SWITCH command for BNJDSERV.
- 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.

Syntax

The DBFULL statement has the following syntax:

DBFULL

►—DBFULL— —nnnnn—►

where:

nnnnn

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

Usage notes

- Code this statement in the member specified by the MEM keyword of the BNJDSERV task statement. The sample member supplied with the NetView program is BNJMBDST.
- To implement the changes, stop and restart the BNJDSERV task. If the changes result in additional extents being used, stop and restart the NetView program.

DEFENTPT

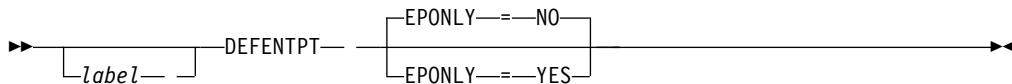
Purpose

The DEFENTPT statement is an optional statement used with the DEFFOCPT statement to specify whether a NetView node is initialized as an operations management focal point or entry point.

Syntax

The DEFENTPT statement has the following syntax:

DEFENTPT



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 allowed 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 notes

- 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.
- 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

Purpose

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.

Syntax

The DEFFOCPT statement has two syntaxes. The syntax you use depends on where the DEFFOCPT statement is located. Refer to *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for examples of how to code the statement. Refer to *IBM Tivoli NetView for z/OS Automation Guide* for a general discussion of NetView focal point and entry point support.

Usage notes

To implement changes to DSICRTTD stop and restart the DSICRTR task. Changes to DSINIT take effect when either of the following conditions occurs:

- An operator at the entry point issues the FOCALPT ACQUIRE DEFFOCP command.
- The DSIDST task is restarted.

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

DEFFOCP Statement in DSICRTTD

Syntax

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

Note: DEFFOCP statements are also residing in DSINIT.

DEFFOCP

```
►-----DEFFOCP---TYPE=---ALERT---,---PRIMARY---nodename-----►  
| [label] |  
|-----►-----  
| ,---BACKUP---nodename2-----►
```

where:

label

Indicates the optional label for the DEFFOCP 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 *xxxxxLUC*, where *xxxxx* 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 *xxxxxLUC*, where *xxxxx* 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.

Usage notes

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.

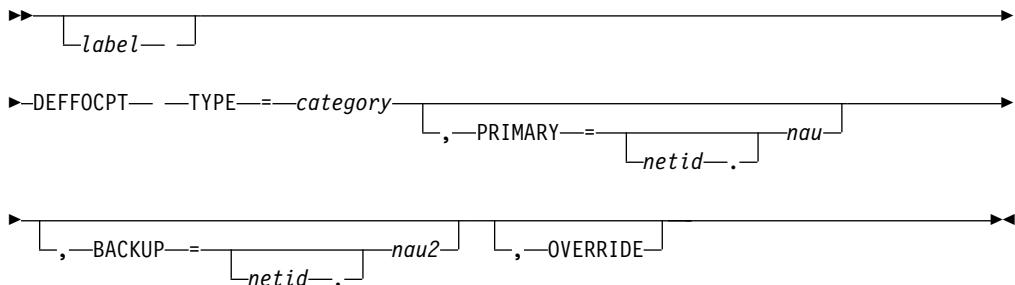
DEFFOCPT Statement in DSINIT

Syntax

Use the following syntax if the DEFFOCPT statements reside in DSINIT. You can specify multiple DEFFOCPT statements for a category; thus, you can 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 DSINIT is used. If you specify more than eight backup focal points, the first eight are used.

Note: A DEFFOCPT statement is also residing in DSICRTTD.

DEFFOCPT2



where:

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. User this operand 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 DSINIT 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 *IBM 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 can be used 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.
- 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 the CNMSTYLE member. If YES is specified, use the CP name from the VTAM program for the focal point name. If NO is specified, use the NetView domain name.

Related statements

DEFENTPT, NPDA.ALERTFWD

DOMAINS**Purpose**

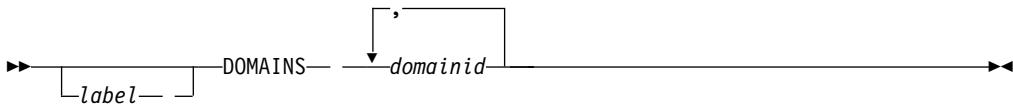
The DOMAINS statement enables the setup of NNT cross-domain communication for operators with specific or general control. This statement shows which NNT

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 373 and “PROFILEN” on page 383 to determine how a PROFILEN statement is associated with an operator.

Syntax

The DOMAINS statement has the following syntax:

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

Purpose

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, the DSIAL2WS task is coded in the CNMSTYLE member in the following way:

```
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 DSRB0=1
DSTINIT XITDI=DSIBKINT

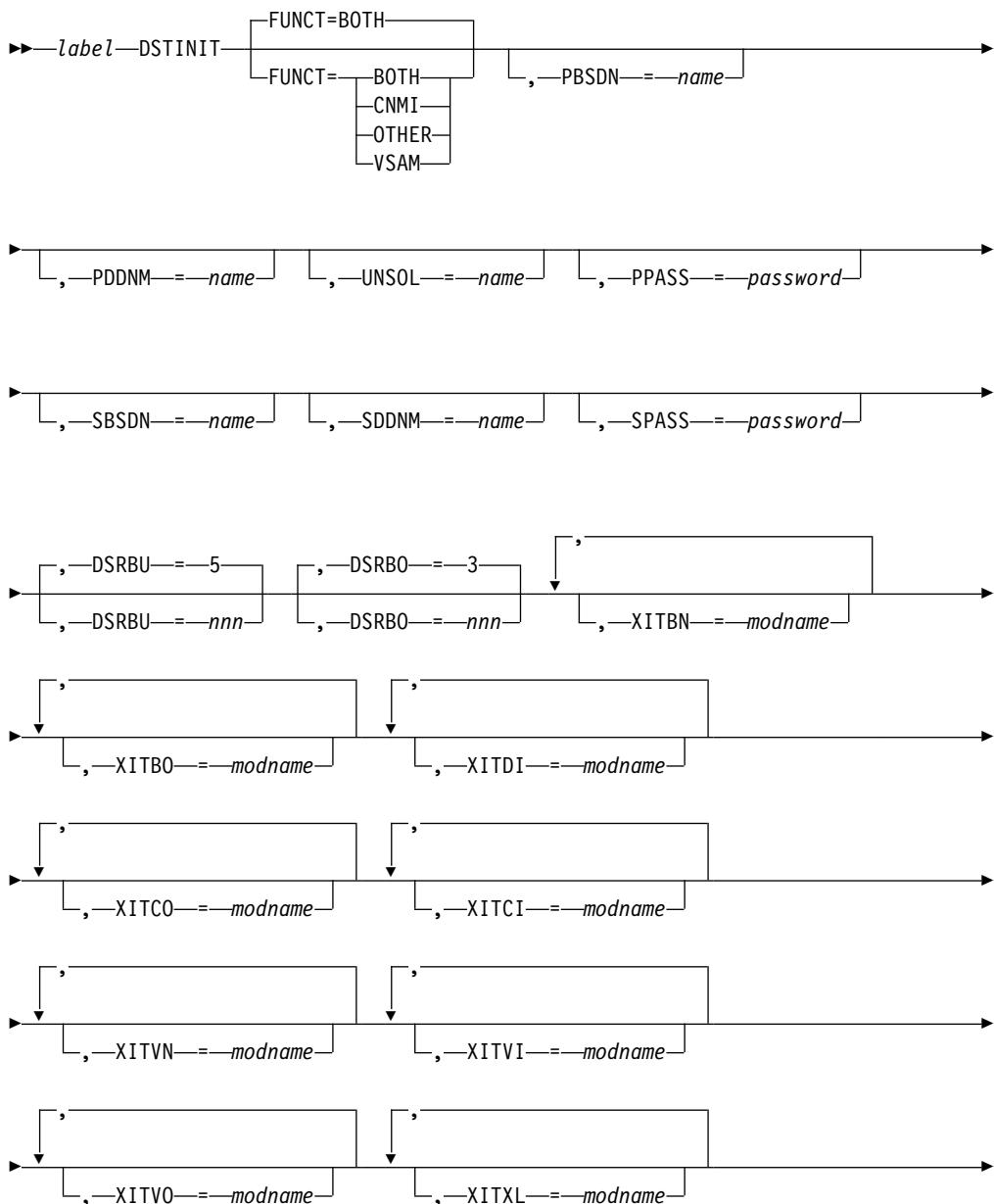
```

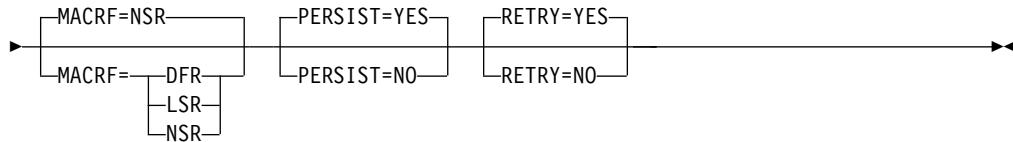
To implement the changes to the member containing the DSTINIT statements stop and restart the associated tasks. If the changes result in additional extents being used, stop and restart the NetView program.

Syntax

The DSTINIT statement has the following syntax:

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.

UNSQL=name

Indicates the 1 - 8 character verb that is the command name of the CMDDEF

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.

SDDNM=*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=*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*|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*|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 DSRBs are available, the requests are queued. The default value is 3, except for the BNJDSERV hardware monitor task. For the BNJDSERV task, the default value coded in the CNMSTYLE member is 5.

The following operands that begin with XIT are 1- through 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:

```
DSTINIT XITCI=MODNAME1, XITCI=MODNAME2, XITCI=MODNAME3
```

If you want to add MODNAME4 and MODNAME5, code the next record in the following way:

```
DSTINIT XITCI=MODNAME4, XITCI=MODNAME5
```

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

XITBN=*modname [, ...]*

Receives control when a sequential log data set is opened for output processing only (not for a resume). Use this exit to put an initialization record into the file.

XITBO=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=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.

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 DSICRT subtask for access to unsolicited CNM data before NetView program internal routing.

XITVN=modname [, . . .]

Receives control when an empty VSAM data set is opened for processing. Use this exit 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 the BNJDSERV hardware monitor task. For the BNJDSERV task, the default coded in the CNMSTYLE member 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 following values are possible:

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 xxxxLUC.

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 the CNMSTYLE member is made every 10 minutes if the first attempt fails. The RETRY setting is controlled by the NLDM.RETRY statement in the CNMSTYLE member.

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, or DSILog, the NetView program sets DSRBU to zero.
- If the task is an LUC, AM/LU, or DSILog, the NetView program sets DSRBO to 1.
- If the task is DSICRTTR (router), the default in the DSICRTTD sample is 5. However, you might need to increase this value as indicated in the sample.
- If the task is CNMTAMEL, set DSRBO to 1.
- Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information about 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 the *IBM 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 login addition to 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

Purpose

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 embedded using the %INCLUDE statement, processing of both the included member and the member that contains the %INCLUDE statement is stopped.

Syntax

The END statement has the following syntax:

END

►►END————►►

Usage notes

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

F (Filter)

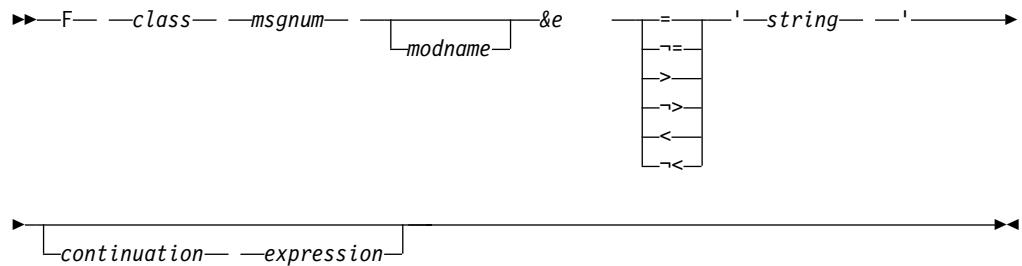
Purpose

Use the F (Filter) statement 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 matches on any name that begins with the characters specified. For example, a test of &2 = "ABCD" matches both 'ABCD' and 'ABCDEF'. You can also use the NETLOG action of the automation table to trigger message indicators. Code the F statement in DSICNM.

Syntax

The F statement has the following syntax:

FILTER



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 called 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 11.

Table 11. Addresses in the Operand List

Word	Address	Description
1	MVTADDR	The MVT address.
2	TXTPTR	The address of the comparison element (<i>&e</i>).

Table 11. Addresses in the Operand List (continued)

Word	Address	Description
3	TXTLEN	The address of a 1-byte field containing the length of &e.
4	RESPTR	The address of the field containing the result of <i>modname</i> processing. (For CNMFANGC, this is one of the node types described in Table 12.)
5	RESLEN	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.)

As an example, CNMFANGC takes &e (a node name) and converts it to one of the 1-character node types as shown in Table 12.

Table 12. 1-Character Node Types

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

This result is then compared to 'string' to determine if an alert is 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 following values are possible:
 - &D Domain identifier.
 - &O Operator identifier.
 - &T Time (in the format *hh:mm:ss*).
 - &M The HDRMTYPE from DSITIB. Refer to *IBM Tivoli NetView for z/OS Programming: Assembler* for more information about the message header type.

&n 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 might not 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.
- To implement the changes to DSICNM, stop and restart the xxxxVMT 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.

Related statements

A

HELD

Purpose

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

Syntax

The HELD statement has the following syntax:

HELD

```
►►HELD— [colorf— ] [attribute— ] [ON— —colorb— ] ►►
```

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 program supplies the following defaults for the HELD statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV

The characters change to reverse video.

UND

The characters are underlined.

- NRM** Normal attributes are used.
- The following operands are intensity operands:

- HIGH** The characters have high intensity.
- LOW** The characters have low intensity.
- The following operand is the alarm operand:

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

HOLDPCNT

Purpose

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.

Syntax

The HOLDPCNT statement has the following syntax:

HOLDPCNT



where:

100 | 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 the panel does not have enough space 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 is displayed 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

Purpose

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

Syntax

The HOLDWARN statement has the following syntax:

HOLDWARN

```
►►HOLDWARN— [colorf— ] [attribute— ] [ON— colorb— ]  
► [NONE] ►
```

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 displaying 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 program supplies the following defaults for the HOLDWARN statement:
 - White foreground
 - Black background
 - Normal highlighting

- High intensity
- Alarm off
- The following operands are the color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

BEEP When a message is shown, an audible alarm sounds.

Related statements

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

IMDAREA

Purpose

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

Syntax

The IMDAREA statement has the following syntax:

IMDAREA

```
►►IMDAREA— [colorf— ] [attribute— ] [ON— colorb— ] ►►
```

where:

colorf

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.
- The NetView program supplies the following defaults for the IMDAREA statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

- UND** The characters are underlined.
- NRM** Normal attributes are used.
- The following operands are intensity operands:
 - HIGH** The characters have high intensity.
 - LOW** The characters have low intensity.
- The following operand is the alarm operand:
 - 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

Purpose

The %INCLUDE statement specifies a tower chain or member (or a file) to be included by the file in which the %INCLUDE statement is used. The tower chain or member that is embedded is treated as if it is a part of the tower chain or member in which it is included. It is embedded at the point at which the %INCLUDE statement occurs.

Syntax

The %INCLUDE statement has the following syntax:

%INCLUDE

```

    %%INCLUDE [tower_chain] -member-
  
```

where:

%INCLUDE

The percent sign (%) must be the first nonblank character on a line, followed by the word INCLUDE. The record cannot be split between lines. INCLUDE must be immediately followed by either a blank space for a *member* specification, or by a plus sign (+) for a *tower_chain* specification. System symbolic characters are supported anywhere following INCLUDE. Refer also to "Examples" on page 335.

tower_chain

If specified, a plus sign (+) must immediately follow INCLUDE (with no blank spaces), followed by the *tower_chain* sequence. A blank space must follow *tower.subtower*, followed by a *member* name. Refer also to "Examples" on page 335.

member

Specifies the member name 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. Refer also to "Examples" on page 335.

Usage notes

- Code %INCLUDE statements in the following items:
 - CNMCMD
 - CNMSCAT2
 - CNMSTYLE
 - DSICNM
 - DSIOPF
 - Automation tables
 - Code point tables
 - Command authorization tables
 - Command facility panel-format definition members, such as CNMSCNFT
 - Hardware monitor (NPDA) members (but not BNJHEADER)
 - Members read using NetView disk services with the INCL option
 - Operator profile (DSIPRF) members
 - Session monitor (NLDM) members
 - Span tables
 - HELPMAP

Note: This is not an exhaustive list of items in which the %INCLUDE statement can be coded.

- 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 *IBM Tivoli NetView for z/OS Programming: REXX and the NetView Command List Language*.

Note:

1. For information about disk services in assembler, see DSIDKS in *IBM Tivoli NetView for z/OS Programming: Assembler*.
 2. For information about disk services in PL/I and C, see CNMMEMR in *IBM Tivoli NetView for z/OS Programming: PL/I and C*.
- When using the %INCLUDE statement in members such as CNMCMD 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, CNMCMD, 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.

Examples

1. The following example specifies the AON.SNA tower chain:

```
%INCLUDE+AON.SNA MEMBER1 MEMBER2
```

Upon finding this statement in a member, the NetView DSIDKS service evaluates the AON.SNA tower chain. If AON.SNA is enabled, processing proceeds as if the statement were %INCLUDE MEMBER1. If AON.SNA is disabled, processing proceeds as if the statement were %INCLUDE MEMBER2.

2. The following example specifies the GRAPHICS tower:

```
%INCLUDE+GRAPHICS MEMBER3 *GRAPHICS not enabled
```

Upon finding this statement in a member, the NetView DSIDKS service evaluates the GRAPHICS tower. If the GRAPHICS tower is enabled, processing proceeds as if the statement were %INCLUDE MEMBER3. If the GRAPHICS tower is disabled, the member is not included and the comment "*GRAPHICS not enabled" is displayed when browsing the CNMSTYLE member and the member in which the statement is coded.

INDENT

Purpose

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

Syntax

The INDENT statement has the following syntax:

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 default that is supplied with the NetView program 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 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.

Related statements

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

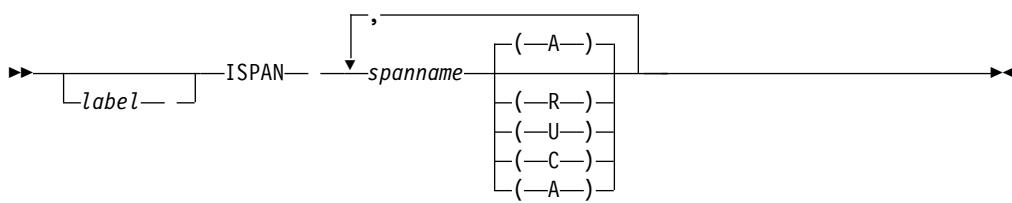
ISPA

Purpose

The ISPA statement sets up the initial span of control for the operator when SECOPTS.OPSPAN = NETV is specified in the CNMSTYLE member 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.

The ISPA statement has the following syntax:

ISPA



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.

Usage notes

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 373 and “PROFILEN” on page 383 to determine how a PROFILEN statement is associated with an operator.

Related statements

AUTH, OPERATOR, SECOPTS, PROFILE, PROFILEN, SPAN

KCLASS

Purpose

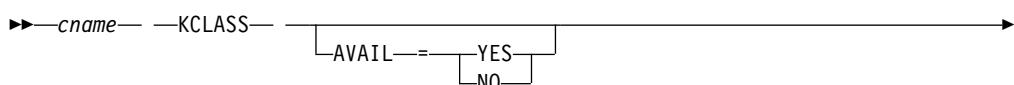
The KCLASS statement defines keep classes. A keep class determines what data is kept for the sessions mapped. A session is mapped into a keep class when the session starts. After a session is mapped into a keep class, it remains there until the session ends.

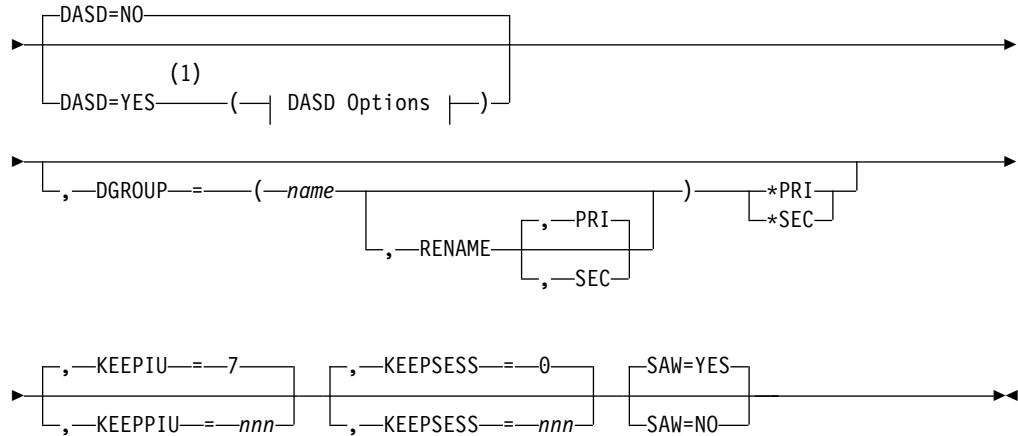
The KCLASS and MAPSESS statements control the amount of session awareness data recorded. If you do not code any of these statements, data for all sessions is recorded according to the defaults for the KCLASS statement.

Syntax

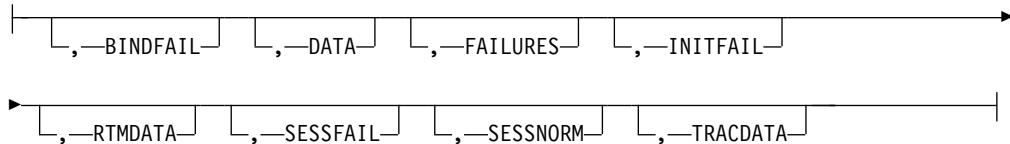
The KCLASS statement has the following syntax:

KCLASS





DASD Options:



Notes:

- 1 Limitations exist 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.

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 NLDM.SESSTATS statement in the CNMSTYLE member. If the value of the NLDM.SESSTATS statement is AVAIL, the default value for AVAIL is YES. If the value of the NLDM.SESSTATS statement is YES or NO, the value for AVAIL is ignored. Table 13 shows combinations of values and their meanings.

Table 13. Combinations of AVAIL and SESSTATS

If SESSTATS=	Then KCLASS AVAIL=	This means that:
AVAIL	Defaults to YES	Availability data is logged.
YES	Is ignored	Both availability data and accounting data are logged.
NO	Is ignored	Availability data or accounting data are not logged.

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 a BIND failure occurs.

DATA

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

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 before BIND flow (INIT failure)

INITFAIL

Records sessions to the VSAM file if a failure to set up occurs before BIND flow (INIT failure).

RTMDATA

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

SESSFAIL

Records sessions to the VSAM file if an abnormal UNBIND reason code exists.

SESSNORM

Records sessions to the VSAM file if a normal UNBIND reason code exists.

TRACDATA

Records sessions to the VSAM file if some trace data associated with the session exists.

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. When 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 are not recorded by DGROUPs. 0 is the default.

SAW=YES|NO

Defines whether session awareness (SAW) data is 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

- 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 the CNMSTYLE member. The KCLASS statements are the first statements in the member.

- Code one KCLASS statement for each keep class you define. You can code as many KCLASS statements as necessary.
- 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 the CNMSTYLE member, the NetView program does not receive session awareness data from VTAM until the ENABLE command is issued.

Related statements

NLDM.SAW, MAPSESS, PCLASS

KEYCLASS

Purpose

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

Related statements

CMDCLASS, OPCLASS, VALCLASS

LASTLINE

Purpose

Use the LASTLINE statement 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.

Syntax

The LASTLINE statement has the following syntax:

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 program supplies the following defaults for the LASTLINE statement:
 - Turquoise foreground
 - Black background
 - Normal highlighting
 - Low intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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

LOCKIND

Related statements

Use the LOCKIND statement 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.

Syntax

The LOCKIND statement has the following syntax:

LOCKIND

```
►►—LOCKIND— — [colorf] — [attribute] — [ON —colorb] —►►
```

where:

colorf

Defines foreground color for the lock indicator. The foreground color must be specified before the background color.

attribute

Defines alarm, intensity, and highlight attributes for the lock indicator. Specify attributes only once for the LOCKIND statement.

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

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 SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The NetView program supplies the following defaults for the LOCKIND statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:
 - BLINK**
 - The characters blink.
 - REV** The characters change to reverse video.
 - UND** The characters are underlined.
 - NRM** Normal attributes are used.
- The following operands are intensity operands:
 - HIGH** The characters have high intensity.
 - LOW** The characters have low intensity.
- The following operand is the alarm operand:
 - BEEP** An audible alarm sounds.

Related statements

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

LOGINIT

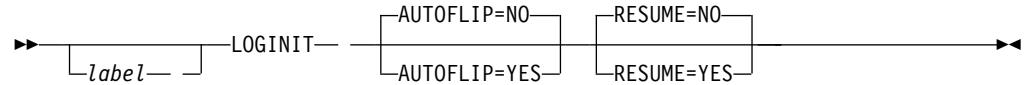
Purpose

The LOGINIT statement defines the initialization operands for the network log and optional NetView sequential logs. The AUTOFLIP operand specifies whether the NetView program automatically continues 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.

Syntax

The LOGINIT statement has the following syntax:

LOGINIT



where:

label

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

AUTOFLIP=NO | YES

Defines whether logging continues on the primary data set when the secondary data set is full. This operand must have one of the following values:

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

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

RESUME=NO | YES

Defines whether logging begins where it left off in either data set, or starts at the beginning of the primary data set. This operand must have one of the following values:

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

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 because of full data sets, depending on where logging left off.

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 statement or by task statements that are using NetView sequential log services. It must follow the DSTINIT statements in the member. The sample member supplied with the NetView program is DSILOGBK.
 - Changes to the member containing the LOGINIT statement do not take effect until you stop and restart the sequential logging tasks, including the DSILOG 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.

Related statements

TASK

LU

Purpose

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.

Syntax

The LU statement has the following syntax:

LU

```
►► [label] LU —targname—,—target—,—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.

target

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

M (Maximum)

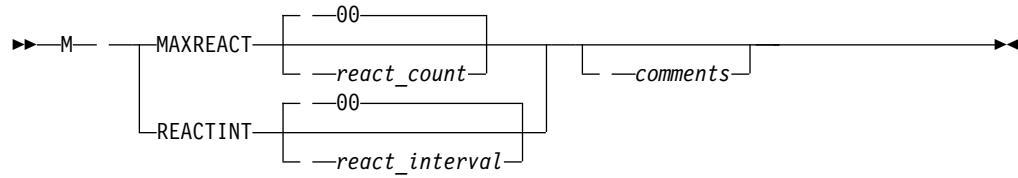
Purpose

Use the M statement 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 are 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.

Syntax

The M statement has the following syntax:

M (Maximum)



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 attempts 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 comments to add as documentation.

Usage notes

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

MAPSESS

Purpose

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 that is named on the NLDM.PERFMEM or NLDM.KEEPMEM statement in the CNMSTYLE member.

Syntax

The MAPSESS statement has the following syntax:

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 primary end point to be specified 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.

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 exist in the position of a ?; that is, A?B is not matched by AB because no character replaces 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.

SEC=*sname*

Names the secondary end point of any session that matches this MAPSESS statement. If the secondary end point to be specified 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 exist in the position of a ?; that is, A?B is not matched by AB because no character replaces 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=nn

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 notes

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

MLINDENT

Purpose

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.

Syntax

The MLINDENT statement has the following syntax:

MLINDENT



where:

0|nn

Defines the number of characters of indentation allowed for each line of a

MLWTO message. The maximum value allowed is 75. The default that is supplied with the NetView program is zero (0).

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.

Related statements

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

MODE

Purpose

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.

Syntax

The MODE statement has the following syntax:

MODE

►— *label* — MODE — *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.

Related statements

ALIASMEM, ORIGNET

MVS

Purpose

You can use the MVS statement to change the color and other attributes of normal messages that were received from MVS when they are displayed on the command facility panel, if the color or attribute was not the default value when it was received from MVS.

Syntax

The MVS statement has the following syntax:

MVS

```
►—MVS [ colorf ] [ attribute ] [ ON colorb ] —►
```

where:

colorf

Defines the foreground color for normal messages that were received from the MVS program that are defined with the color GREEN. The foreground color must be specified before the background color.

attribute

Defines the alarm, intensity, and highlight attributes for normal messages that were received from the MVS program that are defined with the default setting for the attribute that is being defined. You can specify attributes only once for the MVS statement.

ON The color following the ON keyword applies to the background of the messages that are received from the MVS program. This is a required keyword if you specify only a background color.

colorb

Defines the background color for messages received from MVS.

Usage notes

- To create a member containing screen format definitions, use the CNMSCNFT sample member as an example. Specify the name of the member on the SCRNFMT parameter of the DEFAULTS or OVERRIDE command.
- The following default values are initially supplied for the MVS statement:
 - Green foreground
 - Black background
 - Normal highlighting
 - Low intensity
 - Alarm is off
- The following color operands can be specified:
 - BLACK
 - BLUE
 - RED
 - PINK
 - GREEN
 - TURQ
 - YELLOW

- WHITE
- The following highlighting operands can be specified:
 - BLINK**
The characters blink.
 - REV** The characters change to reverse video.
 - UND** The characters are underlined.
 - NRM** Normal attributes are used.
- The alarm operand is BEEP, which indicates that an audible alarm sounds when a message is displayed.

NETCONV_IP

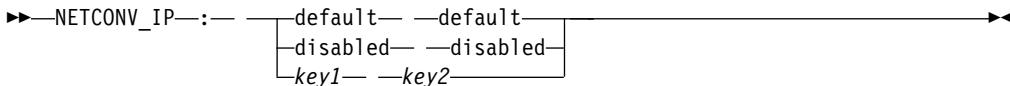
Purpose

The NETCONV_IP statement in the DSITCPRF member, which is in DSIPRF data set, 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, the CNMTAMEL task must be stopped and restarted if it is active. If NETCONV_IP is omitted, a predefined value is used for the encryption keys.

Syntax

The syntax for the NETCONV_IP statement follows:

NETCONV_IP



where:

key1

Specifies the encryption key for the data flowing over the NETCONV TCP/IP sessions to the NetView program (command flow). The length of the key can be 1 - 8 characters. An 8-character key is recommended. Encryption keys can be mixed case, but **default** and **disabled** have special meaning. Choose random printable nonblank characters. Code the encryption keys as 8-character values or 16-hexadecimal digits. If the encryption key is 16 characters and all 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 string..

Specifying **default**, all in lowercase, means that the NetView program 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 that no encryption is provided. 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 the NetView program over the NETCONV TCP/IP sessions. The length of the key can be 1 - 8 characters. An 8-character key is recommended. Encryption keys can be mixed case. Choose random printable nonblank characters.

If you specify default, all in lowercase, the NetView program provides a default encryption key. If *key1* is specified as default, *key2* must be specified as default.

If you specify disabled, all in lowercase, no encryption is provided. If *key1* is specified as disabled, *key2* must be specified as disabled.

Usage notes

- NETCONV_IP must be followed by a colon (:) and any number of blanks.
- Place the DSITCPRF member in a secure (DSIPRF DD) library.
- The NETCONV_IP statement does not apply to NETCONV sessions that use Secure Sockets Layer (SSL).

Related statements

WEB_SERVER

NMCSTATUS (Control File Entry)

Purpose

Use the NMCSTATUS control file entry, or policy definition, to define time schedules for resources in NetView management console views. With these schedules, policy can be applied to NetView management console views to specify when:

- The displayable status of one or more resources in a view is disabled at the NetView management 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 NetView management 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 the NetView program is initialized, define your policy file in the CNMSTUSR or CxxSTGEN member with the following statement:

`POLICY.GRAPHICS = DUIPOLCY`

For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

The DUIPOLCY member in the DSIPARM data set is a sample NMCSTATUS policy file. You can use this file name or create your own DSIPARM member. Refer

to the comments in the CNMSTYLE member for more information on policy definitions.

NMCSTATUS Policy Autotask

If you specified TOWER GRAPHICS in the CNMSTYLE member, the NMCSTATUS policy autotask is started at NetView initialization. Statements are coded in the CNMSTYLE and CNMSTASK members to facilitate the starting of the task and in the DSITBL01 member in the DSIPARM data set to begin processing the NMCSTATUS policy definitions. It is possible to change the name of the DUILPOLI autotask to your own 8-character autotask name. The following CNMSTYLE statement

```
function.autotask.NMCpolicy = DUILPOLI
```

results in the creation of common global variable

```
CNMSTYLE.AUTO.NMCPOLICY = DUILPOLI
```

You can change the autotask name by updating the function.autotask.NMCpolicy statement in the CNMSTUSR or CxxSTGEN member. The CNMSTYLE.AUTO.NMCPOLICY global variable is referenced in the CNMSTASK and DSITBL01 members; do not change these statements.

The RODMname = &CNMRODM statement that is defined in the CNMSTYLE member 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 are shown 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 run command NMCPINIT on autotask DUILPOLI. The NMCPINIT command reads the policy file specified by the POLICY.GRAPHICS statement and determines if the NMCSTATUS policy definitions are valid.

If a DUILPOLCY file is found, the following message is issued indicating NMCSTATUS policy processing has begun:

```
DUI275I BEGIN PROCESSING NMCSTATUS POLICY DEFINITIONS DEFINED IN MEMBER DUILPOLCY
```

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 example shows an error free policy file:

```
DUI250I -----
DUI251I BEGIN ERROR CHECKING FOR MEMBER DUILPOLCY
DUI250I -----
DUI261I NO ERRORS WERE FOUND IN MEMBER DUILPOLCY
DUI250I -----
DUI252I END ERROR CHECKING FOR MEMBER DUILPOLCY
DUI250I -----
```

In addition to the log, the messages can be written to a data set by replacing *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-character 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 *IBM 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 DUIPOLCY
DUI280I -----
DUI283I ALL TIMERS WERE SET SUCCESSFULLY
DUI280I -----
DUI281I END SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLCY
DUI280I -----
```

If a timer cannot 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
CNM249E 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
CNM249E DSISCHED : NO DATA FOUND FOR 'PAYDAY'
DUI280I -----
DUI282I END SETTING TIMERS FOR NMCSTATUS POLICIES DEFINED IN MEMBER DUIPOLCY
DUI280I -----
```

Your timers might 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 example shows 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, cannot occur at the NetView management console.

To update your policy file without recycling the NetView program:

- Update the DSIPARM member defined by POLICY.GRAPHICS in the CNMSTYLE member.
- Issue the POLICY REQ=LOAD command
- Automation on the EZL110I message runs the NMCPINIT command to begin the NMCSTATUS policy processing.

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.

Testing a Policy File

You can test a policy file without starting GMFHS, or RODM, or signing on to NetView management console. With the NMCPTEST command, you can error check NMCSTATUS policy definitions currently loaded in the Policy Repository.

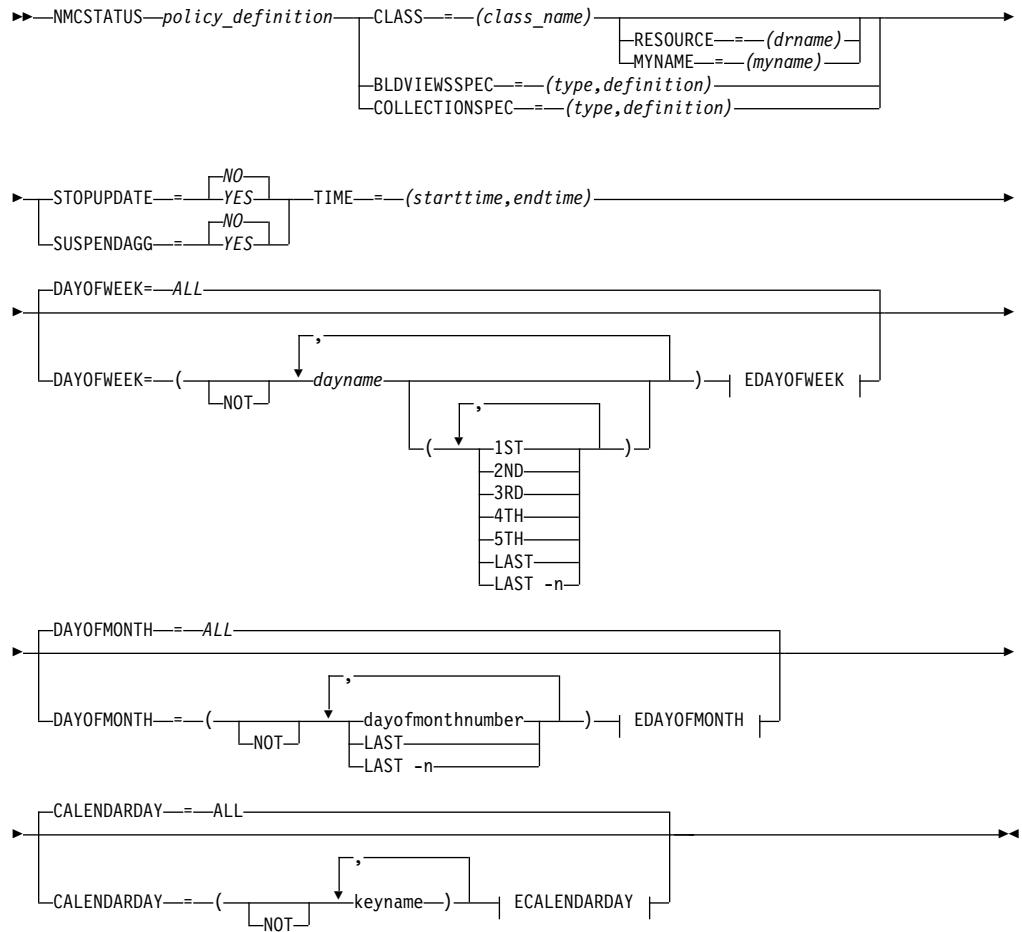
To test your policy file at NetView initialization, follow these steps:

- Update the DSIPARM member defined by POLICY.GRAPHICS in the CNMSTYLE member.
- Update the DSITBL01 member in the DSIPARM data set such that automation on the EZL110I message runs the NMCPTEST command, rather than the NMCPINIT command, 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, issue the POLICY REQ=LOAD command which through automation on message EZL110I runs command NMCPTEST to error check the NMCSTATUS policies.

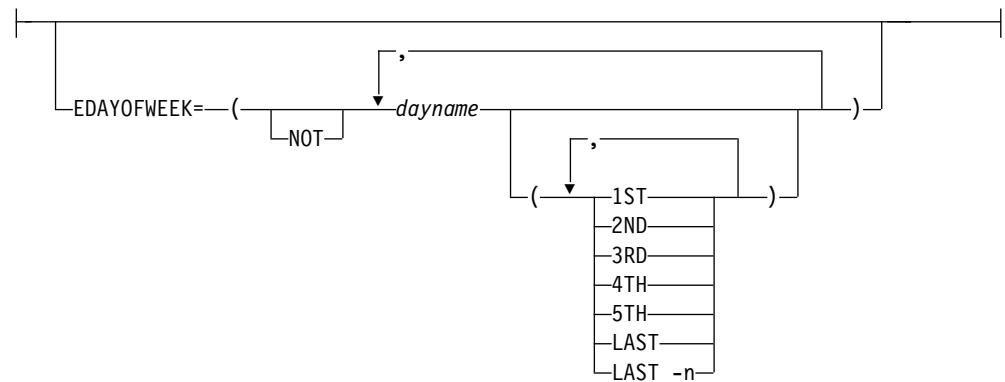
Syntax

NMCSTATUS has the following syntax:

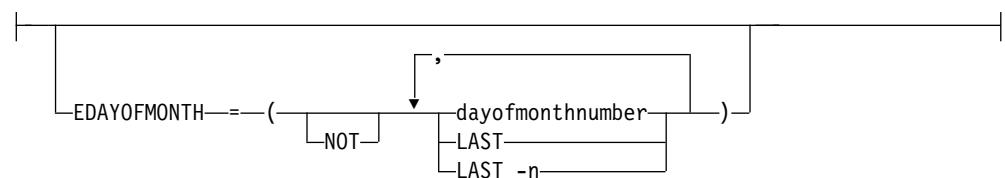
NMCSTATUS



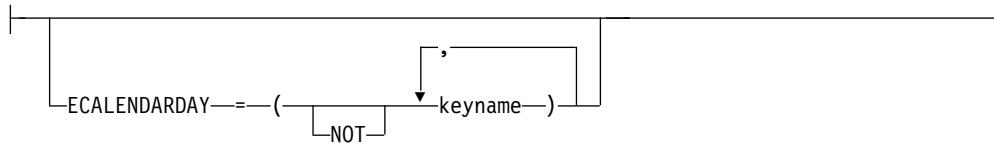
EDAYOFWEEK:



EDAYOFMONTH:



ECALENDARDAY:



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 1 - 32 characters with no embedded blanks, commas, or quotation marks. 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. Wildcard characters are not 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 about class names, refer to the MyName field in the *IBM Tivoli NetView for z/OS Data Model Reference*.

BLDVIEWSSPEC=(*type_definition*)

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(FILE1))`

generates a PIPE command 'PIPE QSAM (DSN) USER.INIT(FILE1)' to read the BLDVIEWS specification. FILE1 contains 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(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 contains 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*)

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(FILE1))
```

generates a PIPE command 'PIPE QSAM (DSN) USER.INIT(FILE1)' to read the RODM Collection Manager (RCM) specification. FILE1 contains 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 contains 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 NetView management console 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 - 23.59.59.

- *endtime*

Defines the ending time when actions are applied to resources in NetView management console 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 - 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)
```

The following values are valid:

- 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)
```

The policy is 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. Unsigned the LAST or LAST-n prevents having to consider the number of a specific weekday within that month. The following values are valid:

- 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. The following values are valid:

- In the range of 1 - 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 DS1656I 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 parentheses.

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 ECALENDARDAY 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 *dname* 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.
- Continued lines must begin after column one.
- Policy_definition must be unique from other NMCSTATUS policy definitions.
- Keywords can be specified in any order.
- Duplicate keywords for policy definitions are not allowed.
- Imbedded blanks are not allowed in the value for keywords BLDVIEWSSPEC, COLLECTIONSPEC, and TIME. For example, TIME=(05.00.00 , 06.00.00) is in error.
- Multiple consecutive commas are not allowed in the value for keywords BLDVIEWSSPEC, COLLECTIONSPEC, TIME, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY or ECALENDARDAY. For example: DAYOFMONTH=(1,2,,4) is in error.
- A comma can not be the first or last character of a value for keywords BLDVIEWSSPEC, COLLECTIONSPEC, TIME, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY or ECALENDARDAY. For example, TIME=(,05.00.00,06.00.00) or DAYOFWEEK=(MON,TUES,) are both in error.
- AON restricts the use of columns 73 to 80 in DSIPARM member DUIPOLCY.

Continuation of NMCSTATUS Policy Definitions

- Continued lines must begin after column one, for example:

```
RESOURCE=
  (ResourceNameThatIsSixtyNineCharactersLong)
```

Do this 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 can not 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 uppercase.
- Values for the following keywords can be specified in mixed case but are interpreted by the NMCPINIT and NMCPTEST commands as uppercase: BLDVIEWSSPEC, COLLECTIONSPEC, SUSPENDAGG, STOPUPDATE, DAYOFWEEK, EDAYOFWEEK, DAYOFMONTH, EDAYOFMONTH, CALENDARDAY, and ECALENDARDAY.
- 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

- 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 not a valid keyword.
- MYNAME is an optional keyword, but if specified, then CLASS is a required keyword and RESOURCE is not a valid 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.

Examples

The following example illustrates how to stop status updates at NetView management console 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.
```

The following example illustrates how to stop status updates at NetView management console 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
```

The following example illustrates how to suspend a group of resources defined by a BLDVIEWSSPEC specification from aggregation on a defined day. PAYDAY is defined in DSIPARM member DSISCHED.

```

NMCSTATUS POLICY3
BLDVIEWSSPEC=(QSAMDSN,USER.INIT(FILE1))
TIME=(00.00.00,24.59.59)
CALENDARDAY=(PAYDAY)
SUSPENDAGG=YES

```

The following example illustrates how to stop updates at NetView management console for a group of resources defined by a BLDVIEWS specification. The updates are stopped on May 1 and started again on June 1. MAY1 and JUNE1 are defined in DSIPARM member DSISCHED.

```

NMCSTATUS POLICY4
BLDVIEWSSPEC=(QSAMDSN,USER.INIT(FILE1))
TIME=(00.00.00,11.59.59)
CALENDARDAY=(MAY1)
ECALENDARDAY=(JUNE1)
STOPUPDATE=YES

```

The following example illustrates how to stop updates at NetView management console 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 POLICY5
BLDVIEWSSPEC=(QSAMDSN,USER.INIT(FILE1))
TIME=(22.00.00,06.00.00)
DAYOFWEEK=(SUN)
EDAYOFWEEK=(MON)
STOPUPDATE=YES

```

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 cannot 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 a NetView management console view, refer to Applying Policy to Views in the *IBM 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 a NetView management console view, refer to Applying Policy to Views in the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

NOPREFIX

Purpose

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.

Syntax

The NOPREFIX statement has the following syntax:

NOPREFIX

►►—NOPREFIX—————►►

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.

Related statements

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

NORMAL

Purpose

Use the NORMAL statement 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.

Syntax

The NORMAL statement has the following syntax:

NORMAL

►►—NORMAL— [colorf] [attribute] [ON—colorb]—————►►

where:

colorf

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.

- ON** Makes the color specified by *colorb* 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 program supplies the following defaults for the NORMAL statement:
 - Turquoise foreground
 - Black background
 - Normal highlighting
 - Low intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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, TITLEOPID,
TITLESTAT, TITLETIME

NORMQMAX

Purpose

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.

Syntax

The NORMQMAX statement has the following syntax:

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 2147483647. 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, TITLEOPID, TITLESTAT, TITLETIME

O MONIT

Purpose

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 in the following ways:

- 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

Syntax

The O MONIT statement has the following syntax:

O MONIT

►—O— —MONIT—————►

where:

O MONIT

Controls automatic node reactivation.

Usage notes

You code the O MONIT option control statement in 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.

Related statements

STATOPT

O RESET

Purpose

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 the perspective of this VTAM node of the status. Since the VTAM node does not know of the existence of this resource, using 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 resolves 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.

Syntax

The O RESET statement has the following syntax:

O RESET

►—O— —RESET—————►

where:

O RESET

Controls the initial status of resources which are known to the status monitor.

Usage notes

Code the O RESET option control statement in the DSICNM member. To implement changes to DSICNM, stop and restart the xxxxVMT task. If changing the member results in additional extents being used, stop and restart the NetView program.

O SECSTAT

Purpose

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.

Syntax

The O SECSTAT statement has the following syntax:

O SECSTAT

►—O— —SECSTAT—————►

where:

O SECSTAT

Designates this status monitor to run as a secondary network resource status monitor.

Usage notes

- Code the O SECSTAT statement in DSICNM. To implement changes to DSICNM, stop and restart the xxxxVMT task. If changing the member results in additional extents being used, stop and restart the NetView program.

- 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 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 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 status updates from VTAM. If you specify O SECSTAT in DSICNM for both NetView programs, neither one gets status updates from VTAM.

O SENDMSG

Purpose

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.

Syntax

The O SENDMSG statement has the following syntax:

O SENDMSG

►—O— —SENDMSG—————►

where:

O SENDMSG

Controls the generation of CNM094I at initialization.

Usage notes

Code the O SENDMSG statement in DSICNM. To implement changes to DSICNM, stop and restart the *xxxxxVMT* task. If changing the member results in additional extents being used, stop and restart the NetView program.

Related statements

SENDMSG

OPCLASS

Purpose

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

Related statements

CMDCLASS, KEYCLASS, VALCLASS

OPERATOR

Purpose

When the SECOPTS.OPERSEC statement is not specified as SAFDEF in the CNMSTYLE member or the OPERSEC parameter on the REFRESH command is not specified as SAFDEF, 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.

Note: In addition to the SAFDEF cases, OPERATOR statements are not needed for operators who will be using a default logon profile when operator security (OPERSEC) is defined as SAFPW or SAFCHECK. See HELP for the LOGPROF keyword of the DEFAULTS command in *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*.

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 the DSIOPF member.

You can dynamically add or delete operators by adding or deleting OPERATOR statements in the DSIOPF member and issuing the REFRESH OPERS command.

Syntax

The OPERATOR statement has the following syntax:

OPERATOR

```
►—opid— —OPERATOR— —PASSWORD—=—password— [—NOCHECK]—►
```

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 the CNMSTYLE member. 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 NetView operator logon without NetView password verification. This option is only valid when SECOPTS.OPERSEC=NETVPW is set in the CNMSTYLE member. 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 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, PROFILEN

ORIGNET

Related statements

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.

Syntax

The ORIGNET statement has the following syntax:

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

Purpose

The PARMSYN statement is obsolete. It is used for migration purposes only. It is replaced by the CMDDEF statement in DSIPARM member CNMCMD. For more information, refer to the *IBM Tivoli NetView for z/OS Installation: Migration Guide*.

Related statements

CMDMDL, CMDSYN

PARTNER

Purpose

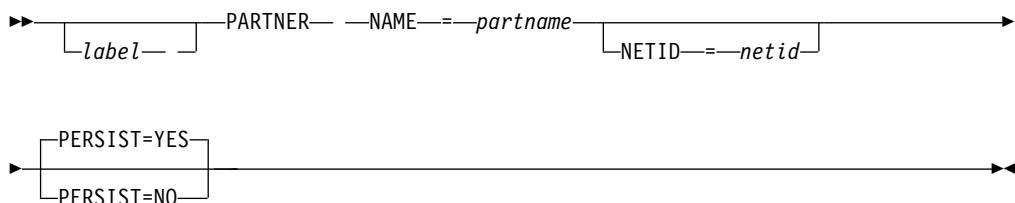
The PARTNER statement specifies whether communications with an LU use persistent conversations. Use this statement only for high performance sessions.

This statement is valid only in DSIHINIT, and it overrides the default value setup by the DSITINIT statement.

Syntax

The PARTNER statement has the following syntax:

PARTNER



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 timeout interval specified in DSICTMOD.

Usage notes

- If two nodes in two networks have the same LU name, VTAM might 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.

PCLASS**Purpose**

The PCLASS statement controls the measurement operands used by the response time monitor. These measurement operands include the following items:

- 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 the CNMSTYLE member.

Code 1 PCLASS definition statement for each performance class that you define. These PCLASS statements are the first statements in the member that is named on the NLDM.PERFMEM statement in the CNMSTYLE member.

If you do not code any of the statements listed in this section, the following defaults are used for all sessions:

- The boundaries between the counters are at 1, 2, 5, and 10 seconds.

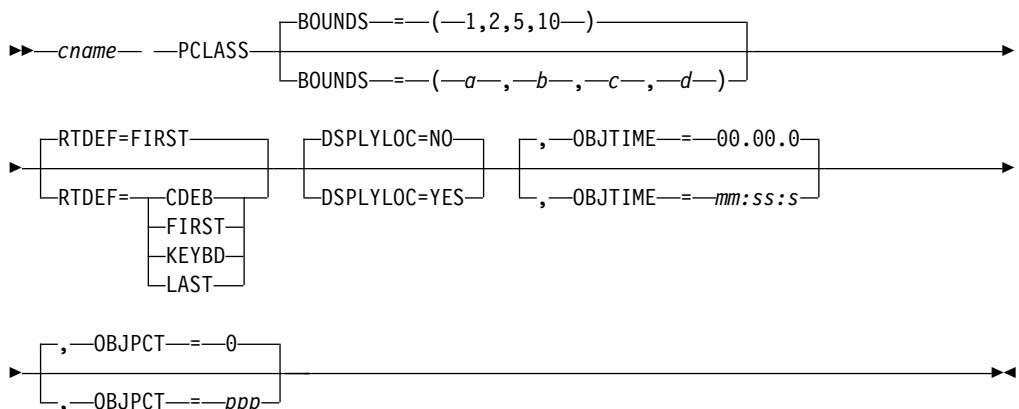
- Response time is defined as *time to first character*.
- You cannot display your own response time.
- No response time objective is defined.

If the defaults are acceptable, you can omit this statement. Do not code the NLD.MPERFMEM statement in the CNMSTYLE member.

Syntax

The PCLASS statement has the following syntax:

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 *cname* in the PCLASS operand. The name starts in column 1 of the statement. The name is required.

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 *a*, the second counter is incremented when response time is greater-than *a* but less-than or equal-to *b*, and so on. The fifth response time counter is incremented when response time is greater-than *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).

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. Use this operand 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.

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.

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 "MAPSESS" on page 347 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=0|ppp

Indicates the percentage portion of the performance objective where:

0 Indicates the default value.

ppp Indicates the percentage of transactions that 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

PORT

Purpose

The PORT statement defines the port for the specified TCP/IP address space on which NetView program waits for connection requests. This statement specifies the NetView program that is to receive a connection request.

The PORT statement is used in the following samples:

- DSIIILGCF
- DSIREXCF
- DSIRSHCF
- DSIRTTTD
- DSITCPCF
- DSIWBMEM
- DUILPFMEM

Syntax

The PORT statement has the following syntax:

PORT

►►PORT—=—*number*—►►

where:

number

Specifies the port number in the range of 1 - 65535.

Usage notes

For the NetView web server, a single blank must precede and follow the equal sign in the PORT statement.

PREFIX

Purpose

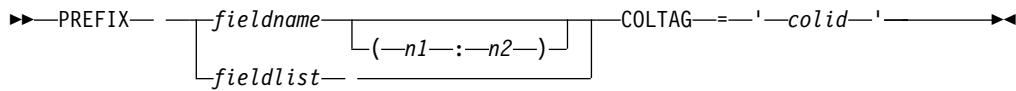
Use the PREFIX statement to define the information that is displayed with a message and the order in which the information is displayed on the command facility panel. The sample member containing the PREFIX statement is CNMSCNFT.

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.

Syntax

The PREFIX statement has the following syntax:

PREFIX



where:

fieldname

Defines the content of a prefix field. The following operands can be used in *fieldname*:

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 2 bytes of user bits formatted in hexadecimal from the IFRAUSR 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 *fieldname* beginning at the fifth character.

fieldlist

Allows up to four mutually exclusive operands to be combined into one field. The operands are separated by logical or (!) 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 *fieldname*. If you do not specify the COLTAG keyword with *fieldlist*, the column tag from the first field listed is used as the default.

COLTAG='colid'

Defines the column identifier. The following values indicate the default column identifiers:

Field Name
Column Tag

DOMID
DOMID

SENDER
SENDER

IFRAUSRC
IFRAUSRC

IFRAUSRB
IFRAUSRB

JOBID
JOBID

JOBNAME
JOBNAME

NMFLAGS
P%

NMTYPE
T

SRCCNETID
SRCCNETID

SRCNAU
SRCNAU

SYSID
SYSID

SESSID
SESSID

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.
- Using each PREFIX statement, you can 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 about the panel. The

defaults that are provided with the NetView program 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, MLINIDENT, NOPREFIX, NORMAL, NORMQMAX, TITLE, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT, TITLETIME

PROFILE

Purpose

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 373 and “PROFILEN” on page 383 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.

Syntax

The PROFILE statement has the following syntax:

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=hclname

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 the CNMSTUSR or CxxSTGEN member. HCL is an optional operand. The IC keyword, when specified, must always be specified as the last keyword. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

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, if the ConsMask CNMSTYLE initialization statement is either not specified or specified with a value of the asterisk (*) character. When ConsMask specifies a value other than *, the value resulting from using ConsMask overrides the value specified with the CONSNAME keyword. 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 about the implications of specifying CONSNAME, refer to the GETCONID and SETCONID command in the NetView online help. The IC keyword, 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 CMDDEF statement as regular (R), both (B), or high (H). The IC keyword, when specified, must always be specified as the last keyword.

Related statements

HARDCOPY, OPERATOR, PROFILEN

PROFILEN

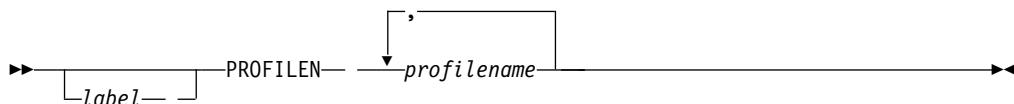
Purpose

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. Operators who will be using a default profile do not need PROFILEN statements; see HELP for the LOGPROF keyword of the DEFAULTS command in *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)*. An OPERATOR statement must precede each PROFILEN statement or group of statements. You code this statement in the DSIOPF member. Profiles are not used when SECOPTS.OPERSEC=SAFDEF is specified in the CNMSTYLE member.

Syntax

The PROFILEN statement has the following syntax:

PROFILEN



where:

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. Multiple profile names must be separated by commas.

Related statements

OPERATOR, PROFILE

PROTOCOL

Purpose

The PROTOCOL statement defines the type of remote IP command server being used.

The PROTOCOL statement is used in the following samples:

- DSIREXCF
- DSIRSHCF

Syntax

The PROTOCOL statement has the following syntax:

►►PROTOCOL = [RExec
RSH]►►

where:

Rexec

Specifies that the DSIRXEXC task is an REXEC server.

Rsh

Specifies that the DSIRSHI task is an RSH server.

Usage notes

- No default value exists if the statement is not present.

REP

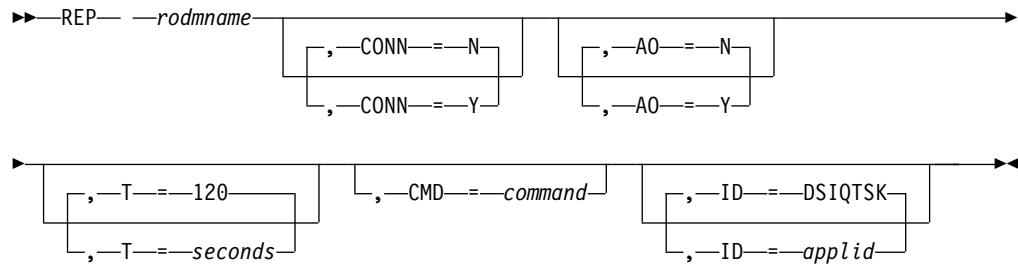
Purpose

Use the REP statement 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.

Syntax

The REP statement has the following syntax:

REP



where:

rodmname

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 requested 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 runtime RODM. You can define only one RODM as the current runtime RODM.

N Specifies that this RODM is not the current runtime RODM. N is the default.

Y Specifies that this RODM is the current runtime 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 called when one of the following conditions is met:

- Connection to RODM occurs.

The command is called with the following parameters:

rodmname CONN

where *rodmname* is the name of the RODM. The command is processed at the first connection only and not on subsequent connection requests when a connection exists.

- A run-time RODM change occurs.

The command is called with the following parameters:

rodmname AUTO

where *rodmname* 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.

applid Specifies the ID of a valid user application. The length of this value cannot exceed eight characters.

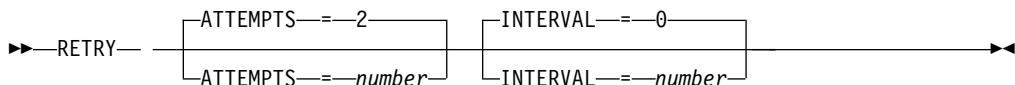
RETRY

Purpose

Use the RETRY statement 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.

Syntax

The RETRY statement has the following syntax:

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 statements

DSI6DST, DSIHPDST

RMTSEC

Purpose

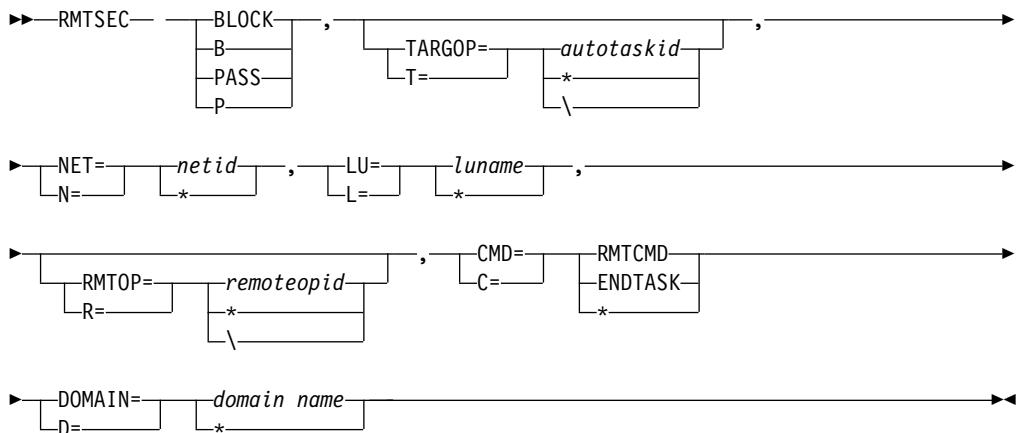
The RMTSEC statement authorizes or blocks remote operators from using the RMTCMD or ENDTASK commands to start or stop autotasks on a distributed NetView program. The distributed NetView host adds RMTSEC statements in the table that is used for RMTCMD security verification for remote operators, nodes, or LUs that operators want to authorize or block. The sample that is provided with the NetView program for the RMTCMD security validation table is DSISECUR.

For more information on using this table for RMTCMD and ENDTASK security, see *IBM Tivoli NetView for z/OS Security Reference*.

Syntax

The RMTSEC statement has the following syntax:

RMTSEC



where:

BLOCK|B

Specifies that the request is denied.

PASS|P

Specifies that the request is 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 is responsible for the RMTCMD or ENDTASK request.

LU|L=Luname|*

Specifies the domain identifier of the remote operator that is responsible for the RMTCMD or ENDTASK request.

RMTOP|R=remoteopid|*|\\"

Specifies the user ID of the remote operator that is responsible for the

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 is responsible for the RMTCMD or ENDTASK request. This keyword cannot 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

- The operator ID, network identifier, and domain identifier of the responsible operator is determined based on the following values of the RMTAUTH security setting, which is set by either the SECOPTS.RMTAUTH statement in the CNMSTYLE member or the REFRESH RMTAUTH command:
 - If the RMTAUTH value is SENDER, the operator that issued the incoming RMTCMD or ENDTASK request is used to compare with the NET, LU, and RMTOP values.
 - If the RMTAUTH value is ORIGIN, the operator that originated the request, which eventually resulted in the RMTCMD or ENDTASK request that was received at this domain, is used to compare with the NET, LU, and RMTOP values.
- You can use a question mark (?) to replace one character as a wildcard character, for example:

```
RMTSEC P TARGOP=O??R* NET=NETA LU=C???A RMTOP=NETOP? CMD=*
```

The command has the following results:

- TARGOP=O??R* finds OPER1 and OPER22, but it does not find OPR1.
- NET=NETA finds only NETA.
- LU=C???A finds C123A and CIBMA, but it does not find C34A or CIBMAA.
- RMTOP=NETOP? finds NETOP1 and NETOPA, but it does not find NETOP or NETOP22.
- CMD=* finds 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.

SENDMSG

Purpose

Use the SENDMSG statement 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.

Syntax

The SENDMSG statement has the following syntax:

SENDMSG

►—SENDMSG— —*resource_type*—————►

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.

Usage notes

If you request additional information for a type of resource and your network contains a large number of such resources, the status monitor 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

Purpose

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.

The SOCKETS statement is used in the following samples:

- DSIILGCF
- DSIREXCF
- DSIRSHCF
- DSIRTTD
- DSITCPCF
- DSIWBMEM
- DSIZCECF
- DUIFPMEM

Syntax

The SOCKETS statement has the following syntax:

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.

SPAN

Purpose

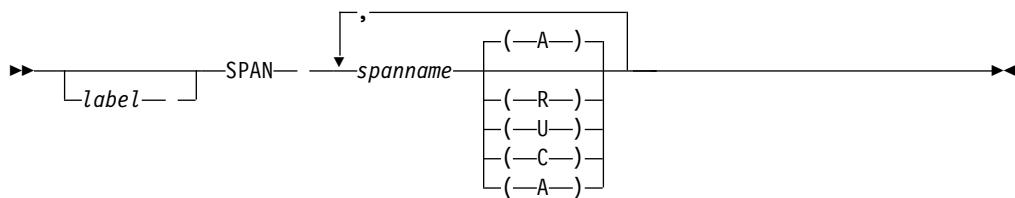
Use the SPAN statement for span of control when SECOPTS.OPSPAN=NETV is specified in the CNMSTYLE member or OPSPAN=NETV is specified on the REFRESH command. SPAN is coded in an operator profile and defines the spans that an operator can add to span of control by using the START SPAN command. These spans are not in the operator's control at logon.

You can use SPAN as often as necessary to define all the span names you want. 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 373 and "PROFILEN" on page 383 to determine how a PROFILEN statement is associated with an operator.

Syntax

The SPAN statement has the following syntax:

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 multi-write access to resources and views defined to *spanname*. This is the default if no access level is specified.

Related statements

AUTH, ISPAN, OPERATOR, PROFILEN

STATOPT

Purpose

Use the STATOPT operand values to control the functions of the status monitor. 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 take the following actions:

- 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 following basic types of STATOPT statements can be used within major node definition statement members:

- A statement following an initial major node macroinstruction statement (LBUILD, VBUILD, or BUILD macroinstruction) that defines a descriptive name for the major node when you do not want to use the default.
- 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.

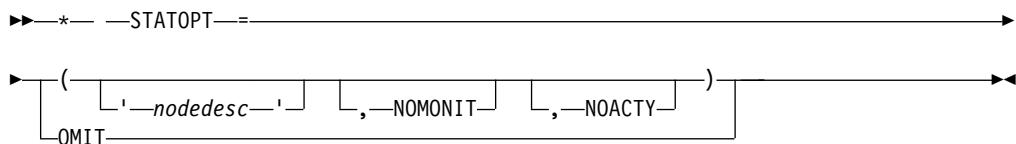
If NCP001 is activated first, then deactivated, and then NCP002 is activated, the status monitor might not display the correct status for either of the LU1 resources. If NCP002 is then deactivated and NCP001 is activated again, the status monitor displays the correct status for the LU1 under NCP001.

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.

Syntax

The STATOPT statement has the following syntax:

STATOPT



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.

Use the following rules for adding the STATOPT statement:

- 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 *nodedesc* 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 *nodedesc* 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 14 are used.

Table 14. STATOPT Default Descriptions

Node Group	Default Description
Application major node	APPL MAJ NODE
Application minor node	APPLICATION
CDRM major node	CDRM MAJ NODE
CDRM minor node	CDRM
CDRSC major node	CDRSC MAJ NODE
CDRSC minor node	CDRSC
Local non-SNA terminal major node	LOCAL 3270 MAJ
Local non-SNA terminal minor node	LOCAL dev-addr
Local SNA PU/LU major node	LCL SNA MAJOR
Local SNA PU minor node	LOCAL SNA PU
Local SNA LU minor node	LOCAL SNA LU
NCP major node	NCP MAJOR NODE
Line minor node	LINE
Remote PU minor node	PU
Remote LU minor node	LU
Remote cluster minor node	CLUSTER
Remote terminal minor node	TERMINAL

Table 14. STATOPT Default Descriptions (continued)

Node Group	Default Description
Channel-channel major node	CA MAJOR NODE
Line minor node	LINE
PU minor node	PU
LU minor node	LU
CLUSTER minor node	CLUSTER
TERMINAL minor node	TERMINAL
Switched major node	SWITCHED MAJOR
Switched PU minor node	SWITCHED PU
Switched LU minor node	SWITCHED LU
LAN major node	LAN major node
Line minor node	LINE
PU minor node	PU
PACKET major node	PACKET major node
LINE minor node	Line
PU minor node	PU
LU minor node	LU

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 370 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 notes

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

SUBSYSTEM

Purpose

The SUBSYSTEM statement defines the DB2 subsystem to which the NetView program connects. This statement is contained in member DSIDB2DF and is used by the DSIDB2MT task.

Syntax

The SUBSYSTEM statement has the following syntax:

SUBSYSTEM

►►SUBSYSTEM— ——*subsystemname*————►►

where:

subsystemname

Name of the DB2 subsystem to which NetView connects. Subsystemname is a character value with a length of 1 - 4 characters.

TASK (DSIQTSKI)

Purpose

Use the TASK statement to define a NetView autotask or operator task that the NetView program-to-program interface command receiver uses to process commands. You can define as many as 64 tasks to the program-to-program interface command receiver by coding multiple TASK statements in DSIQTSKI.

Syntax

The TASK statement has the following syntax:

TASK

►►TASK— —*taskname*————►►

where:

taskname

Specifies a valid NetView task name that can be 1 - 8 characters.

TCPANAME

Purpose

Use the TCPANAME statement to specify the name of the TCP/IP address space.

The TCPANAME statement is used in the following samples:

- DSIILGCF

- DSIREXCF
- DSIRSHCF
- DSIRTTTD
- DSITCPCF
- DSIWBMEM
- DSIZCECF
- DUILPMEM
- DUILIGHB

Syntax

The TCPANAME statement has the following syntax:

TCPANAME

►►TCPANAME— = —*proc*—►►

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 notes

For the NetView web server, a single blank must precede and follow the equal sign in the TCPANAME statement.

TITLE

Purpose

Use the TITLE statement to change the color of the title line displayed on the command facility panel. The sample member containing the TITLE statement is CNMSCNFT. You can code the TITLE statement only once.

Syntax

The TITLE statement has the following syntax:

TITLE

►►TITLE— [*colorf*] [*attribute*] [ON— *colorb*] ►►

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 specified by *colorb* 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 program supplies the following defaults for the TITLE statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

BEEP An audible alarm sounds.

Related statements

ACTION, CMDLINE, COLUMNHEAD, HELD, HOLDPCNT, HOLDWARN,
IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL,

NORMQMAX, PREFIX, TITLEDATE, TITLEDOMID, TITLEOPID, TITLESTAT,
TITLETIME

TITLEDATE

Purpose

Use the TITLEDATE statement to change the color of the date on the title line displayed on the command facility panel. The sample member containing the TITLEDATE statement is CNMSCNFT. Code the TITLEDATE statement only once.

Syntax

The TITLEDATE statement has the following syntax:

TITLEDATE

```
►—TITLEDATE— — [colorf] — [attribute] — [ON — colorb] —►
```

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 TITLEDATE statement.

ON Makes the color specified by *colorb* 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 program supplies the following defaults for the TITLEDATE statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.**UND** The characters are underlined.**NRM** Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.**LOW** The characters have low intensity.

- The following operand is the alarm operand:

BEEP An audible alarm sounds.**Related statements**

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

TITLEDOMID**Purpose**

Use the TITLEDOMID statement to change the color of the NetView program domain name displayed on the command facility panel. The sample member containing the TITLEDOMID statement is CNMSCNFT. You can code the TITLEDOMID statement only once.

Syntax

The TITLEDOMID statement has the following syntax:

TITLEDOMID

►►—TITLEDOMID— [colorf] [attribute] [ON —colorb] ►►

where:

colorf

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 specified by *colorb* 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 program supplies the following defaults for the TITLEDOMID statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

BEEP An audible alarm sounds.

Related statements

ACTION, CMDLINE, COLUMNHEAD, DATE, HELD, HOLDPCNT, HOLDWARN, IMDAREA, INDENT, LASTLINE, LOCKIND, MLINDENT, NOPREFIX, NORMAL, NORMQMAX, PREFIX, TITLE, TITLEDATE, TITLEOPID, TITLESTAT, TITLETIME

TITLEOPID

Purpose

Use the TITLEOPID statement to change the color of the NetView operator identifier displayed on the command facility panel. The sample member containing the TITLEOPID statement is CNMSCNFT. You can code the TITLEOPID statement only once.

Syntax

The TITLEOPID statement has the following syntax:

TITLEOPID

```
►►—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 specified by *colorb* 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 program supplies the following defaults for the TITLEOPID statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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, TITLESTAT,
TITLETIME

TITLESTAT**Purpose**

Use the TITLESTAT statement to change the color of the status indicators at the end of the title line displayed on the command facility panel. The sample member containing the TITLESTAT statement is CNMSCNFT. You can code the TITLESTAT statement only once.

Syntax

The TITLESTAT statement has the following syntax:

TITLESTAT

```
►►—TITLESTAT— [colorf] [attribute] ON— —colorb— ►►
```

where:

colorf

Defines foreground color for the status indicators. You must specify the foreground color before the background color.

attribute

Defines alarm, intensity, and highlight attributes for the status indicators. You can specify attributes only once for the TITLESTAT statement.

ON Makes the color specified by *colorb* apply to the background of the status indicators. This is a required keyword if you specify only a background color.

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 NetView program supplies the following defaults for the TITLESTAT statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operands:

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 following operands are highlight operands:

BLINK

The characters blink.

REV

The characters change to reverse video.

- UND** The characters are underlined.
- NRM** Normal attributes are used.
- The following operands are intensity operands:
 - HIGH** The characters have high intensity.
 - LOW** The characters have low intensity.
 - The following operand is the alarm operand:
 - 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

Purpose

Use the TITLETIME statement to change the color of the time on the title line displayed on the command facility panel. The sample member containing the TITLETIME statement is CNMSCNFT. You can code the TITLETIME statement only once.

Syntax

The TITLETIME statement has the following syntax:

TITLETIME

```

    ➤—TITLETIME— — [colorf] — [attribute] — [ON — colorb] — ➤
  
```

The diagram shows the syntax of the TITLETIME statement. It starts with a double arrow pointing right, followed by the keyword 'TITLETIME'. Then there is a horizontal line with three square brackets. The first bracket contains the variable 'colorf', the second contains 'attribute', and the third contains 'ON' followed by 'colorb'. Another horizontal line with a double arrow points to the right from the end of 'colorb'.

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 specified by *colorb* 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 program supplies the following defaults for the TITLETIME statement:
 - White foreground
 - Black background
 - Normal highlighting
 - High intensity
 - Alarm off
- The following operands are color operand:

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 following operands are highlight operands:

BLINK

The characters blink.

REV The characters change to reverse video.

UND The characters are underlined.

NRM Normal attributes are used.

- The following operands are intensity operands:

HIGH The characters have high intensity.

LOW The characters have low intensity.

- The following operand is the alarm operand:

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

VALCLASS

Purpose

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

Related statements

CMDCLASS, KEYCLASS, OPCCLASS

VPDINIT

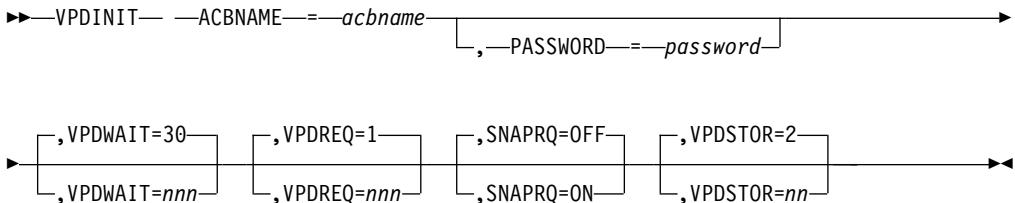
Purpose

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.

Syntax

The VPDINIT statement has the following syntax:

VPDINIT



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 waits 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.

SNAPRQ=OFF | ON

Specifies whether the SNAP trace option is 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.

VPDST0R=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.

No standard way is available to determine the optimal value for VPDSTOR. If you receive message DWO019 (insufficient storage), increase the value until you no longer receive the message.

WEB SERVER

Purpose

The WEB_SERVER statement in the DSITCPRF member, which is in DSIPRF data set, defines the encryption keys for HTTP server sessions. To change the encryption keys on the HTTP server sessions, the DSIWBTSK task must be stopped and restarted if it is active.

Syntax

The syntax for the WEB_SERVER statement follows:

WEB SERVER

where:

key1

Specifies the encryption key for the data flowing over the HTTP server sessions to the NetView program (command flow). The length of the key can be 1 - 8 characters. An 8-character key is recommended. Encryption keys can be mixed case, but default and disabled have special meaning. Choose random printable nonblank characters. Code the encryption keys as 8-character values or 16-hexadecimal digits. If the encryption key is 16 characters and all 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 string..

Specifying default, all in lowercase, means that the NetView program 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 that no encryption is provided. 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 the NetView program over the HTTP server sessions. The length of the key can be 1 - 8 characters.

An 8-character key is recommended. Encryption keys can be mixed case. Choose random printable nonblank characters.

If you specify `default`, all in lowercase, the NetView program provides a default encryption key. If `key1` is specified as `default`, `key2` must be specified as `default`.

If you specify `disabled`, all in lowercase, no encryption is provided. If `key1` is specified as `disabled`, `key2` must be specified as `disabled`.

Usage notes

- `WEB_SERVER` must be followed by a colon (:) and any number of blanks.
- Place the DSITCPRF member in a secure (DSIPRF DD) library.

Related statements

`NETCONV_IP`

WINDOW

Purpose

The WINDOW display panel performs certain actions when an operator types EXEC (with a subcommand) on the WINDOW display. By default, the provided CNMEXEC sample is called to implement the EXEC function. If you want to have a different REXX procedure called by the EXEC action, specify the name of that procedure on the following statement.

See sample CNMEXEC for options and coding suggestions.

Syntax

The WINDOW statement has the following syntax:

WINDOW

►►—WINDOW.EXEC— = —CNMEXEC—►►

where:

CNMEXEC

Specifies the name of the EXEC function to be performed. The default is the provided CNMEXEC sample.

Usage notes

This function is disabled if your system has a command definition for EXEC.

Chapter 4. Policy File Definitions

This chapter describes the policy definitions that you customize for:

- Base NetView functions
- IP management functions
- Base AON functions
- SNA and TCP automation components

Syntax Usage Note: Policy file definitions are keyword oriented. Definitions must start in column 1. Start continuation lines for a definition in column 2 or beyond.

Policy 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 the following information:

- Resource monitoring
- Environment specifications
- Resource threshold values
- Automation modes
- Message forwarding
- Automation operators
- Notification operators
- Notification policy

This section is a reference for system programmers coding the automation policy in the control file. The following types of control file entries are used:

- 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 alphabetic 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

Purpose

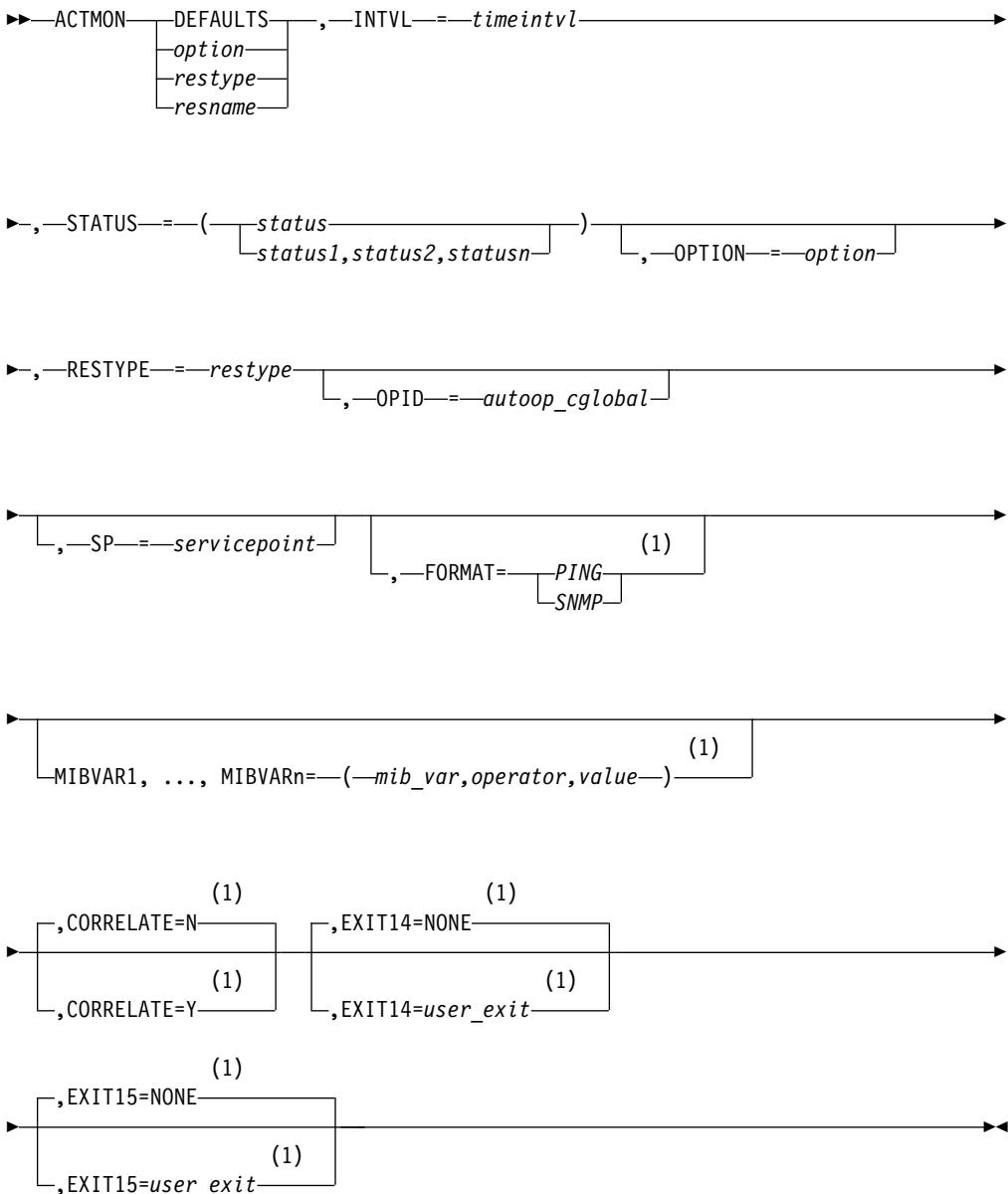
The ACTMON control file entry applies to IP management and AON functions.

The ACTMON control file entry defines active monitoring policy for critical resources in the network. When a resource is actively being monitored, the NetView program sets a timer to check the status of the resource. If the resource is not active, the NetView program initiates failure processing (also known as

recovery monitoring) and suspends active monitoring in favor of recovery monitoring. Once the resource becomes active, the NetView program resumes active monitoring.

Syntax

ACTMON



Notes:

- 1 This parameter is valid only when OPTION=IP390 is specified.

where:

DEFAULTS | *option* | *restype* | *resname*

DEFAULTS

Required if you are going to perform any active monitoring. This

statement defines active monitoring definitions for any component. 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 might 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 component resource types as indicated in the *IBM Tivoli NetView for z/OS 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. AON does not support resources with an IP Version 6-format address.

INTVL

Required at some level. This value might be defined differently for each resource you want to monitor, or might 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*). 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. See Table 15 for examples of the STATUS specification.

Table 15. Status specification examples

STATUS specification	Search
ACTIV	Explicit status match of ACTIV
ACT*	Match that begins with ACT
/ACTIV	All statuses except ACTIV
/ACT*	All statuses except those beginning with ACT

OPTION

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.

Do not use OPTION if you specify ACTMON DEFAULTS.

RESTYPE

Indicates the 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 EZLRT statement of the AON option definition table (EZLTABLE, FKVTABLE, or FKXTABLE).

Do not use RESTYPE if you specify ACTMON DEFAULTS or ACTMON *option*.

OPID

Optional. This value might be defined differently for each resource to be monitored, or might 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 a single AUTOOP statement has two operator IDs, 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 or IP management. Defines which 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 TCP390 definition that matches the value in this parameter. See “TCP390” on page 476 for more information about TCP390 definitions. You can define this value differently for each resource you want to monitor, or you can define it at any nonspecific level definition (DEFAULTS, TCP390, or *resource_type*).

FORMAT

Specifies one of the following options to be used to determine the status of the resource.

PING Pings the resource to check its status.

SNMP

Uses SNMP MIB polling to check the status of the resource.

MIBVAR1 – n

Multiple MIBVAR statements can be defined and are required only for SNMP thresholding (requires FORMAT=SNMP).

CORRELATE

Set to Y for trap correlation of IPHOST, IPROUTER, or IPTN3270 and the respective interface. Set the CORRELATE parameter to Y for resources only when SNMP is available. All known interfaces are used to correlate the status. Only use correlation for interfaces installed on critical IPHOSTs or IPROUTERS. N is the default.

EXIT14

Defines additional processing of SNMP interface table. The default value is NONE. Called for FORMAT=SNMP.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE.

Usage notes

- The ACTMON statement requires that the IPMGT or AON tower be enabled.
- The active monitoring function checks the availability of a router's IP address and the IP address of its links (interfaces). If the NetView program cannot ping a link, it starts failure processing for that link. The NetView program also puts the router in a LINKDOWN status, but active monitoring continues.
- The active monitoring function causes both the host name and IP address of a name server to be pinged. If the host name cannot be pinged, the name server is not operational and the NetView program starts failure processing.

Examples

- The following example shows the ACTMON DEFAULTS control file entry that is required for the NetView program to perform 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 the NetView program. 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*).

- The following example shows 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.

- 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
```

- 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
```

- 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, ...

```

- 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

```

- 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

Purpose

The ADJNETV control file entry applies to base AON functions.

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.

Syntax

```

-->--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.

- Domains that you specify as adjacent NetView programs must be physically next 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 pass through domain CNM02. However, if domain CNM02 is inactive, domain CNM99 is the alternate domain for passing the information to domain CNM03.

AUTOOPS

Purpose

The AUTOOPS control file entry applies to IP management and AON functions.

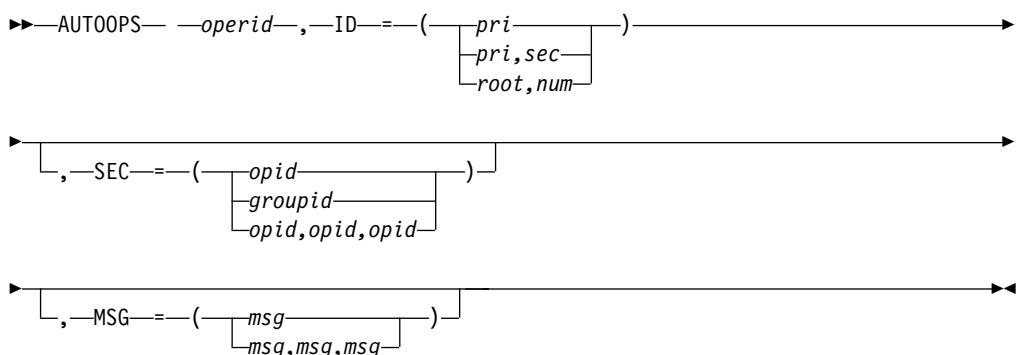
The AUTOOPS control file entries define NetView operator IDs that function as automation operators. Automation operators are NetView tasks that respond to network messages without requiring operator intervention. Each automation operator initiates actions through procedures defined in the control file.

Automation operators are assigned specific messages to act upon. Messages are assigned on the MSG parameter of the AUTOOPS entry.

The NetView program defines multiple automation operators. Each operator processes as a separate task within the NetView program. This task design permits the NetView program to distribute its workload among the automation operators, thereby improving system performance.

Syntax

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.

NetView 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 might 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 at least two common global variables:

- automation operator name
- automation operator name with the number 2 appended at the end

Examples of automation operator names are MSGOPER and MSGOPER2. If only one ID is defined, both common global variables contain the same name and ID.

The following automation operator names are provided:

AIPOPER (AON)

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 (AON, IPMGT)

Sends alerts and resolutions to NetView over an LU 6.2 session.

BASEOPER (AON, IPMGT)

Provides backup for other automation operators.

CONNOPER (AON)

The automation operator used for active connection monitoring.

DVIPOPER

Used for DVIPA polling processes.

GATOPER (AON)

The outbound gateway operator for automation notification forwarding.

INFOPER (AON)

Serializes the updates to the inform log.

MSGOPER (AON, IPMGT)

Formats and issues notifications and DDF updates.

NETOPER (AON/SNA)

Initiates routines based on the NetView automation table and AON/SNA generic failure and recovery routines.

OIVOPER (AON)

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 (AON/TCP, IPMGT)

Used for TCP/IP automation

TRAPOPER (AON)

Used for trap automation processes

WKSTOPER (AON/TCP)

Sends and receives commands and responses between AON and a workstation with the interface installed.

X25OPER

Used by X25 automation processes (AON/SNA only)

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 the NetView program 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.

The NetView program only supports offloading work of NETOPER, MSGOPER, TCPOPER, and AIPOPER. Provide at least two IDs for the automation operator definitions. Code a root autotask name to be used along with a number to identify how many autotasks are needed. For example, to define five AIPOPER autotasks (AUTAIPI through AUTAIPI5), code the following command:

```
AUTOOPS AIPOPER, ID=(AUTAIPI,5)
```

This creates five common global variables: AIPOPER, AIPOPER2, AIPOPER3, AIPOPER4, and AIPOPER5.

Note: When coding any AUTOOPS definitions, follow the syntax provided in the sample policy definitions shipped with the NetView program.

The outbound gateway operator is defined as a NetView operator. The default naming convention for the operator ID is the three letter prefix GAT, followed by the *domain-ID*.

You can change the autotask names as needed for your installation.

SEC

Specifies one or more NetView operator IDs to receive a copy of the messages defined for this automation operator. Messages that are sent to these operator IDs are not subject to automation. .

You can also specify a valid NetView operator 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. By defining a group name, you can send a secondary message to all operators belonging to the group. Define groups in the CNMSTUSR or CxxSTGEN member by using the ASSIGN.groupname.GROUP statement. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

MSG

Specifies the messages to be routed to this automation operator. The NetView ASSIGN statement is used 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

- The AUTOOPS statement requires that the IPMGT tower or the AON tower be enabled.
- NetView message assignment statements are issued based on the order of the automation operators in the control file. The assignment statements must not overlap and must be placed in order from most specific to least specific.
- Define at least one automation operator, referred to as the base operator (BASEOPER), with an AUTOOPS entry. Several automation operators are provided for optimum operation. 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 abnormally ends, 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.

If you change the primary assignments provided with the NetView program, results are unpredictable.
- In situations where a command list runs for a long time period, 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 (indicated by DS1008I message), the NetView program restarts the task. The NetView program builds and initiates the AUTOTASK command based on the DS1008I message.

Examples

The following examples of AUTOOPS control file entries are SNA examples.

- The following example shows 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 DS1008I 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.
- The following examples are of additional automation operator definitions by AUTOOPS:

```
AUTOOPS BASEOPER, ID=AONBASE
AUTOOPS MSGOPER, ID=(AONMSG,2), MSG=EZL*
```

- 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.

- 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 autotask names used are AONMSG1 through AONMSG9. AON contains 10 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 autotask definitions.

- In the following example, the operator ID that is used for automatic deletion of resources from the Operator Intervention network View (OIV) is defined:

```
AUTOOPS OIVOPER, ID=AUTOIV1
```

A TIMER statement is required to activate the automatic deletion.

CDLOG

Purpose

The CDLOG control file entry applies to base AON functions.

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 known to 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 across multiple domains, define RMTCMD sessions for each GATOPER with CDLOG definitions.

Syntax

CDLOG

```
►—CDLOG— —operid.domainid—,—SESSTYPE—=—NNT—
                                         | RMT |
                                         |
                                         , TARGPW=PROMPT—
                                         , TARGPW=pswd—
                                         , INIT=N—
                                         , INIT=Y—
```

► [,—DESC—=*user_description*] ►

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.

Note: Information on password checking can be found in *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

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.

Usage notes

- Separate each parameter by a comma and follow all control file entry guidelines. To use CDLOG to log on to cross-domains, remove all DSI809A statements from CNMCM. 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 the CNMSTUSR or CxxSTGEN member for cross-domain logon. It is not necessary to define RMTCMD sessions in the CNMSTUSR or CxxSTGEN member. For information about changing CNMSTYLE statements, see *IBM Tivoli NetView for z/OS Installation: Getting Started*.

- 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

Does not establish the cross-domain session when OPER1 logs on to CNM01.

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.

CPCPSESS (SNA)

Purpose

The CPCPSESS control file entry applies to base AON functions.

The CPCPSESS control file entry identifies Advanced Peer-to-Peer Networking control-point to control-point sessions that are to be actively monitored by AON/SNA.

Syntax

```
►—CPCPSESS— —alias—,—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=USIBMTA.TA01
CPCPSESS CPCPS2,CP1=USIBMTA.TA1CP208,CP2=USIBMTA.TA01
CPCPSESS NOWAY,CP1=WAYNE.GARTH,CP2=USIBMTA.TA01
```

DDFGENERIC

Purpose

The DDFGENERIC control file entry applies to base AON functions.

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.

This statement applies to base AON functions.

Syntax

DDFGENERIC

```
►►DDFGENERIC [func.] field,--VALUE==(--val1,val2,...valn--)►►
   [ ,--OTHER==value] [ ,--LEVELUP==tree_level]
```

where:

func.

Defines the function. The period is required to separate the *func* and *field* parameters.

field

Specifies the field to be compared.

VALUE

Defines the field values that match this DDFGENERIC statement.

OTHER

Specifies the default value used if the LEVELUP parameter or value is not found.

LEVELUP

Specifies the EZLTREE level at which to display the resources matching this DDFGENERIC statement.

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  
      6 CDRSC  
      6 PU  
      6 LU  
      6 SESSION  
      6 APPL  
      6 ERR  
    5 APPN  
      6 CP  
      6 EN  
    5 X25  
      6 X25MCH  
      6 X25PU  
 4 TCPIP  
    5 IP390  
    6 SP  
    6 HOST  
    6 ROUTER  
    6 INFC  
    6 LINK  
    6 NAMESERV  
    6 PORT  
    6 TN3270
```

- 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 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

Purpose

The DDFGROUP control file entry applies to base AON functions.

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.

Syntax

```
►►DDFGENERIC— —groupname— ,—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 notes

The group name is in the EZLTREE member and is on a DDF panel.

Examples

```
DDFGROUP SANFRAN,LIST=(SASF*,GWATSF,GWSFSD)
```

ENVIRON AIP

Purpose

The ENVIRON AIP control file entry applies to base AON functions.

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.

Syntax

ENVIRON AIP

```
►►ENVIRON AIP— [LINEPORT=N] [PULINKSTA=N] [CDRM=N] [NCP=N]►►
```

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.

Examples

- The following settings are the default settings shipped in the ENVIRON AIP control file:

```
ENVIRON AIP,LINEPORT=Y,
          PULINKSTA=Y,
          CDRM=Y,
          NCP=Y
```

- The following example show how you can extend the resources available for AIP processing.

```
ENVIRON AIP,LINEPORT=Y,
          PULINKSTA=Y,
          CDRM=Y,
          NCP=Y,
          AS400=Y
```

Additional user-written customization is required to set AIP operator status for AS400.

ENVIRON DDF

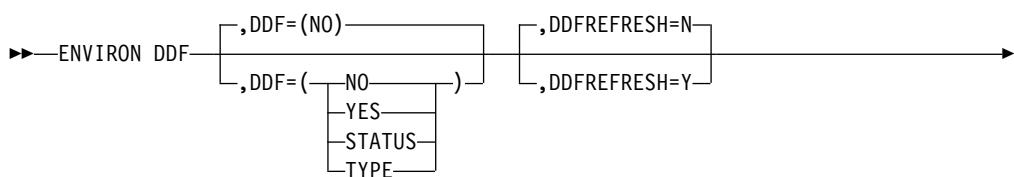
Purpose

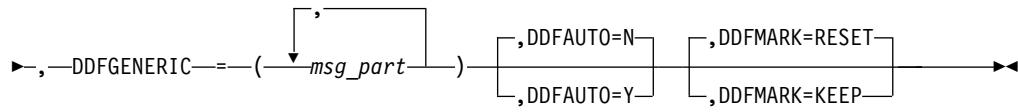
The ENVIRON DDR control file entry applies to base AON functions.

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 422).

Syntax

ENVIRON DDF





DDF

Specifies how the display is to be updated. You can use the following values 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 can 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 that 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 - 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 DDF descriptors can be saved under multiple generic values. (See "DDFGENERIC" on page 422).

Define at least one DDFGENERIC statement 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 is to be saved, when the resource in the DDF is updated, if the operator is not logged on. You can use the following values 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 is to be saved regardless of the status of the operator that marked the resource.

Examples

The following example specifies that the colors are determined by status, the DDF program is not to run, and the resource type and service point parts of the message are 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 is lost.

ENVIRON EXIT

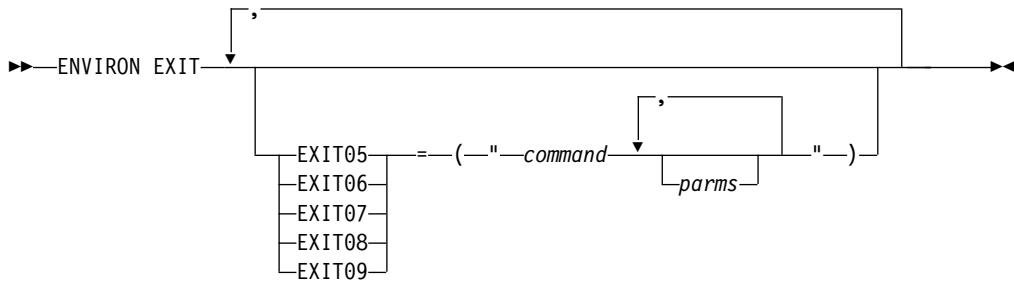
Purpose

The ENVIRON EXIT control file entry applies to base AON functions.

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.

Syntax

ENVIRON EXIT



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 an SNA resource, or class of resources by setting the TGLOBALs documented. EXIT09 can be used to perform a particular function 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 notes

- You can code multiple exit lines, such as EXIT05, as shown in the examples. When multiple EXIT0*n* 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
EZLEZLEAGRNRNC

EZLEATHR
EZLEZLEATHRRNC

EZLECAUT
EZLEZLECAUTRC

- You must set the return code to a numeric value 0 - 9. If the GLOBAL contains anything else, the routine returns a return code of 99.

ENVIRON RACF

Purpose

The ENVIRON RACF control file entry applies to base AON functions.

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.

Syntax

```
►--ENVIRON RACF-->
  [ ,--OWNER==owner_domain-- ]
  [ ,--SHARE==(share_domain,share_domain,share_domain)-- ]
  [ ,--LIST==(owner_domain,share_domain,share_domain)-- ]
  [ ,--MASK==%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. Information about password checking can be found in *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

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 8-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 in the following way:

- 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.

Examples

- 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.

```

ENVIRON RACF,OWNER=CNM01,
      SHARE=(CNM02,CNM04,CNM99),
      MASK=%X%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.

- 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.

```

ENVIRON RACF,LIST=(CNM01,CNM02,CNM03,CNM99),
      LIST=(CNM11,CNM12,CNM13,CNM19),
      LIST=(CNM21,CNM22,CNM23,CNM29)

```

ENVIRON SETUP

Purpose

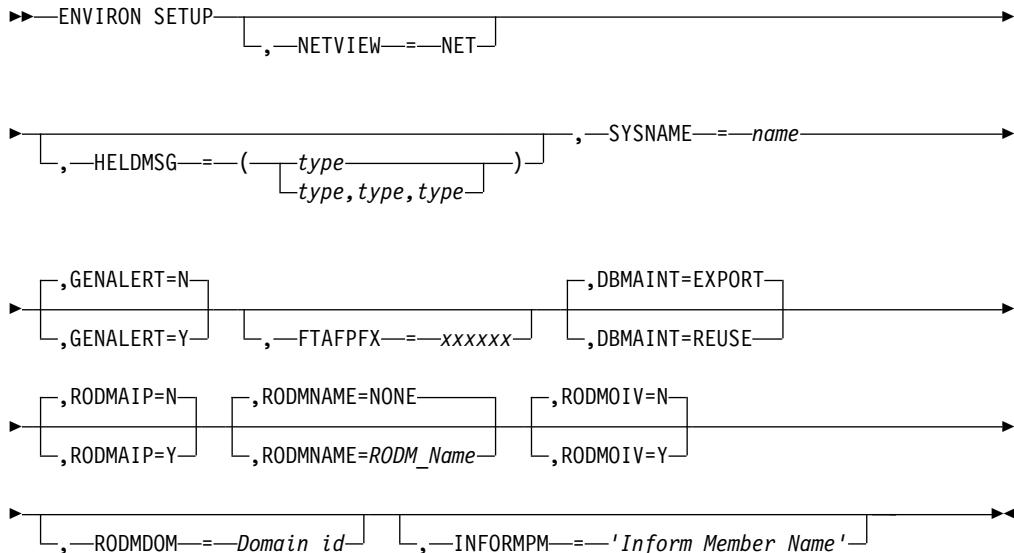
The ENVIRON SETUP control file entry applies to base AON functions.

The ENVIRON SETUP control file entry identifies attributes of the NetView program that is running AON and its supporting operating system. Use the ENVIRON SETUP entry to tailor the control file.

The ENVIRON SETUP entry is optional. If you do not include it, AON uses all of the defaults.

Syntax

ENVIRON SETUP



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 following message types are used:

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.

SYSCNAME

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 (domain ID of this NetView program) 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 and the Resource Object Data Manager (RODM) with automation information, code GENALERT=Y.

FTAFFPFX

Defines the first six characters of the full-screen TAF Access Method Control Blocks (ACBs) defined in the NetView AON ACB definitions. FTAFFPFX 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 TAF*nnn*, where *nnn* is the last 3 characters of the domain ID.

The FTAFFPFX parameter is required only if you are using the FULLSESS entry. The default is TAFxxx (last 3 characters of the domain ID).

- When FTAFFPFX 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 FTAFFPFX is defined, code the following TAF ACB:

xxxxxxyy

where:

xxxxxx Value specified in FTAFFPFX 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: 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 AONAIP command in the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* for information regarding the RODM user ID.

RODMOIV

Defines whether resources are added to or deleted from the OIV. RODMAIP must be enabled and RODMNAME defined for the RODMOIV setting of YES to be effective.

RODMDOM

If the domain where RODM and NetView management console 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 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 are misdirected.
- The Operator Intervention View (OIV) is created when you specify RODMOIV=YES and RODMAIP=YES in the ENVIRON SETUP control file entry. AON attempts to create the view during initialization, or later, when AON attempts to populate the view.
- AIP and OIV functions use the RODM user ID domainid concatenated with AON when accessing RODM.

Examples

The following example shows an ENVIRON SETUP control file entry:

```
ENVIRON SETUP,NETVIEW=NET,  
GENALERT=YES,  
FTAFFX=TAFA1F,  
SYSNAME=NTV6D,
```

```

DBMAINT=EXPORT,
RODMAIP=YES,
RODMOIV=YES,
RODMNAME=RODM1,
HELDMSG=(INFO,WARN,ERROR,ACTION)

```

ENVIRON TIMEOUT

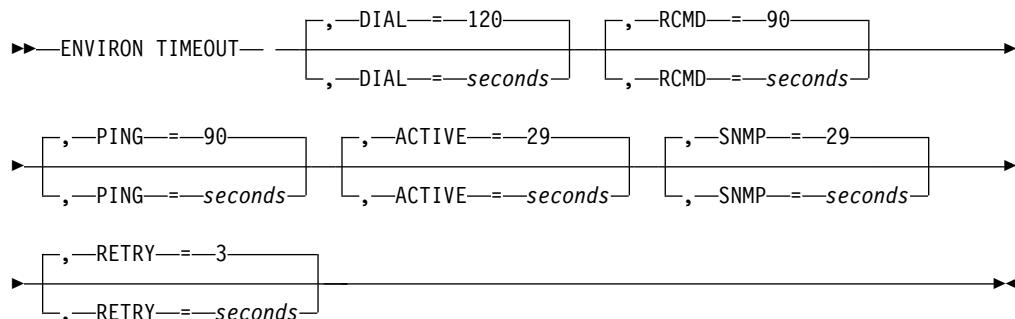
Purpose

The ENVIRON TIMEOUT control file entry applies to base AON functions.

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.

Syntax



DIAL

Specifies the time to wait before a DIAL attempt times out.

RCMD

Specifies the time to wait before a RUNCMD times out.

PING

Specifies the timeout parameter on PING commands.

ACTIVE

Specifies the time to wait before a resource activation attempt times out.

SNMP

Specifies the time to wait before SNMP commands timeout.

RETRY

Specifies the wait time in seconds between PING RETRY attempts.

Usage notes

- AON adds a suffix of TIME to all keyword names and creates CGLOBAL names for them. The following names are created by AON and its components:
 - DIALTIME
 - RCMDTIME
 - PINGTIME
 - ACTIVETIME

- SNMPTIME
- RETRYTIME
- 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 6 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, the NetView program can echo warnings that cannot be suppressed about the WAIT statement to the NetView console. This does not impact AON operation.

Examples

- 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 5 seconds for VTAM to respond to a request to activate. All other specifications are defaults.

- The following AON entry has 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 can cause the active monitoring process to slow down because of excessive wait time. Active monitoring performs 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

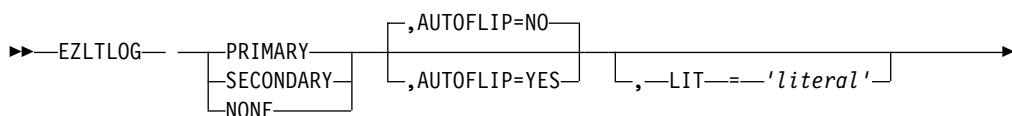
Purpose

The EZLTLOG control file entry applies to base AON functions.

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.

Syntax

EZLTLOG



```
▶ [ ,--JOB=--cmd ] ▶
```

{PRIMARY|SECONDARY|NONE}

Specifies whether this is the primary or secondary log. If NONE is specified, EZLTLOG is not initialized and no NLOG logging is performed. Specifying NONE might 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 are supplied.
- The AON sample library contains the procedure and the IDCAMS control statements necessary to perform switch and offload. Jobs EZLSJ007 and EZLSJ009 copy the VSAM data sets into a sequential history log and clears the AON logs. When the logs are clear, the NetView program 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:

```
EZLTLOG PRIMARY,JOB=USER.PROCLIB(EZLSJ007),AUTOFLIP=YES,  
        LIT='AUTOMATION LOG'
```

```
EZLTLOG SECONDARY,JOB=USER.PROCLIB(EZLSJ009),AUTOFLIP=YES,  
        LIT='AUTOMATION LOG'
```

- 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.

```
EZLTLOG PRIMARY,AUTOFLIP=YES  
EZLTLOG SECONDARY,AUTOFLIP=YES
```

FORWARD FOCALPT

Purpose

The FORWARD FOCALPT control file entry applies to base AON functions.

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.

Syntax

```
►—FORWARD FOCALPT—,—PRI—=—domain—  
                                                  |  
                                                  , —BKUP—=—domain—►
```

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, the NetView domain name where AON resides is the default. If AON defaults, it considers this NetView program 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.

Examples

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 438.

FULLSESS

Purpose

The FORWARD FOCALPT control file entry applies to base AON functions.

The FULLSESS control file entry identifies applications with which AON operators can establish full-screen AONTAF sessions automatically using the operator interface.

Syntax

```
►►FULLSESS— —name—,—APPLID==appl— [—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 FTAFFPX 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

*xxxxyy**nnnn*, where *xxxxyy* is the 5-character domain ID and *nnnn* is the ACB number (000–099)

TAF logon ACB name

TAFyyFnn, where *yy* is the last 2 characters of the domain ID and *nn* is the ACB number (00–99)

- Ensure that the number of TAF ACBs equals the number of NetView operators.

Examples

The following example shows a full-screen entry for TSO.

```
FULLSESS TSO,APPLID=TA1N1
```

GATEWAY

Purpose

The GATEWAY control file entry applies to base AON functions.

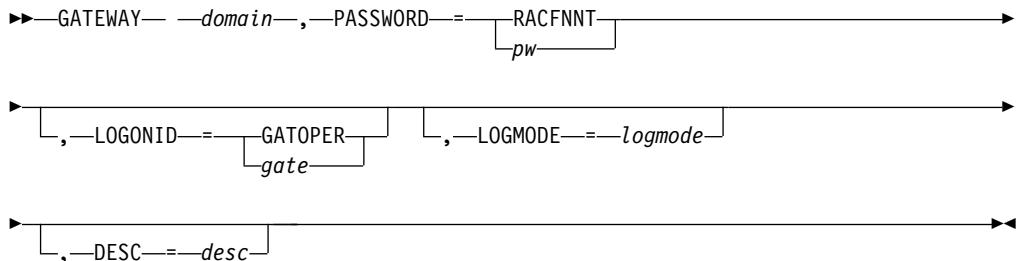
The GATEWAY control file entry identifies NetView domains to which AON routes automation notification messages. When automation notification forwarding is used, this control file entry is defined at the distributed hosts.

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 436.)

Gateway entries must also be defined in the control file of the focal point so that the focal point can reestablish 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 supported by message routing.

Syntax



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.

Note: Information on password checking can be found in *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

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, customize your GATEWAY policy definitions. Only one product can use *GATdomain* as the gateway ID; therefore, to customize the AON gateway task, choose a unique gateway ID for each GATEWAY LOGINID=*gate*, where *gate* is the gateway ID.

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 the NetView program 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 436.

Examples

- The following example defines a gateway entry without RACF password protection:

```
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.

- The following example defines a gateway entry with RACF password protection:

```
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.

- 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.

```
GATEWAY CNM03,DESC='DIST. HOST',PASSWORD=PWD3
```

The following entry is the distributed entry:

```
GATEWAY CNM01,DESC='DIST. HOST',PASSWORD=PWD1
```

NetView domain CNM01 is the usual focal point to which automation notifications are forwarded.

- The following example defines an intermediate host (CNM02), and distributed host (CNM03). The primary focal point entry points to the intermediate host.

```
GATEWAY CNM02,PASSWORD=PWD2  
ADJNETV CNM03,DOMAIN=CNM02
```

The following entry is the intermediate host entry:

```
GATEWAY CNM01,PASSWORD=PWD1  
GATEWAY CNM03,PASSWORD=PWD3
```

The following entry is the distributed host entry:

```
GATEWAY CNM02,PASSWORD=PWD3  
ADJNETV CNM01,DOMAIN=CNM02
```

All automation notifications from the distributed host are forwarded to CNM01 through CNM02, which is defined as an adjacent NetView host. A backup focal point has not been defined for the distributed host.

ADJNETV entries are explained in “ADJNETV” on page 414

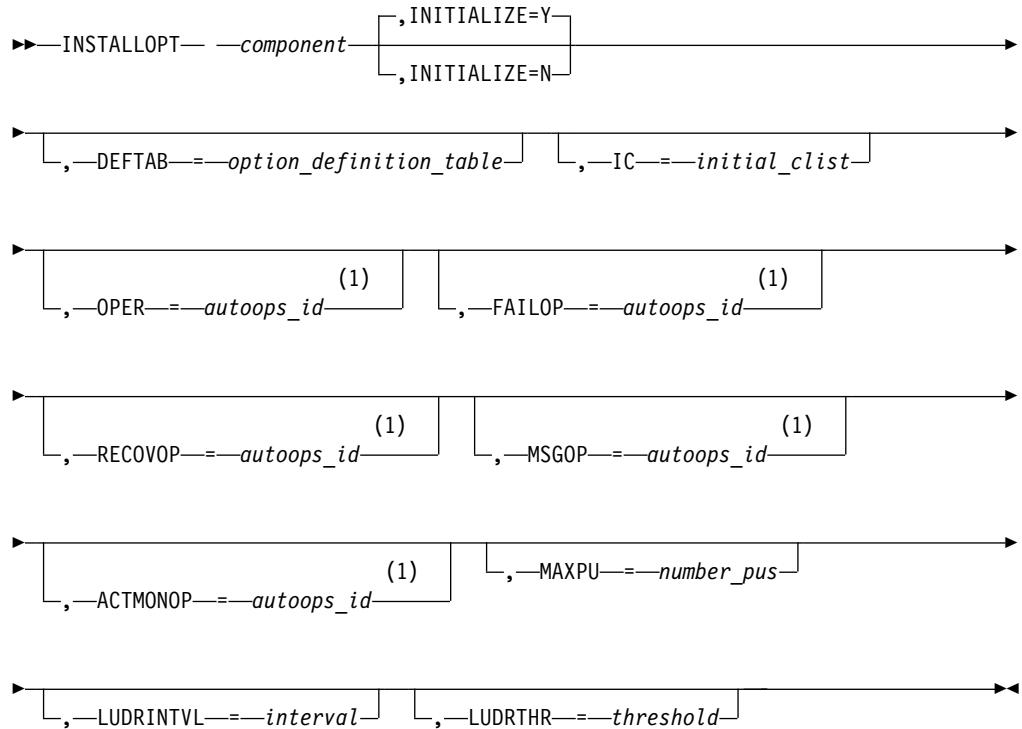
INSTALLOPT

Purpose

The INSTALLOPT control file entry applies to base AON functions.

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.

Purpose



Notes:

- 1 This keyword value is the value of the common global variable created by an AUTOOPS statement.

component

Defines the component installed, for example, specify AON for AON:

- AON
- APPN
- IP390
- SA
- SNA
- 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. Ensure that an AUTOOPS statement exists 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.

ACTIONOP

Directs active monitoring to run under a particular automated operation. Define an AUTOOPS statement for this operator ID.

MAXPU

Specifies the maximum number of switched PUs available for use by X25 processing.

LUDRINTVL

Specifies the default interval for the LUDRSTAT command.

LUDRTHR

Specifies the default threshold for the LUDRSTAT command.

Usage notes

To install and initialize automation, define an INSTALLOPT statement.

Examples

- The following default entry uses the option definition table EZLTABLE to start AON:

```
INSTALLOPT AON,INITIALIZE=Y,DEFTAB=EZLTABLE
```

- 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
```

- The following example defines the MVS TCP/IP function. Installop uses the existing Installop parameters

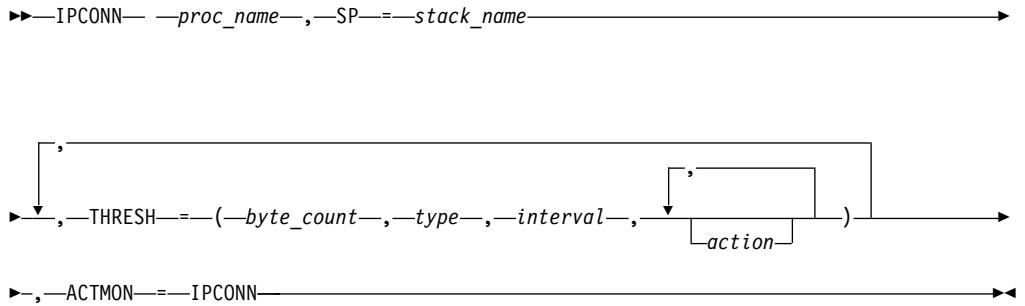
```
INSTALLOPT IP390,INITIALIZE=Y,FAILOP=TCP30P2,RECOVOP=TCP30P2,MSGOP=TCP30PM
```

IPCONN**Purpose**

The IPCONN control file entry applies to IP management and AON/TCP functions.

The IPCONN control file entry defines thresholds to be used when monitoring connections for a given application.

Syntax



proc_name

The name of the procedure associated with the application. Wild card characters (*) are supported.

stack_name

The name of the stack to use to monitor connections with this application. Ensure that this matches 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

The following action types are valid:

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 notification policy.

DROP Breaks the connection and logs the appropriate message if NOTIFY is not specified.

Note: The *action* specification is optional. You can specify multiple actions. If the same action is specified more than once (for example, NOTIFY, NOTIFY), the first specification is processed and duplicate specifications are ignored.

ACTMON=IPCONN

This is required to reference the timer interval to prevent more than one timer for all IPConn definitions.

Usage notes

- The IPCONN statement requires that the IPMGT or AON/TCP tower be enabled.
 - 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 uses the value specified for *action*.

Examples

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 actions occur:

- If idle for more than four minutes, the connection is 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 is broken and message FKF108I is issued.

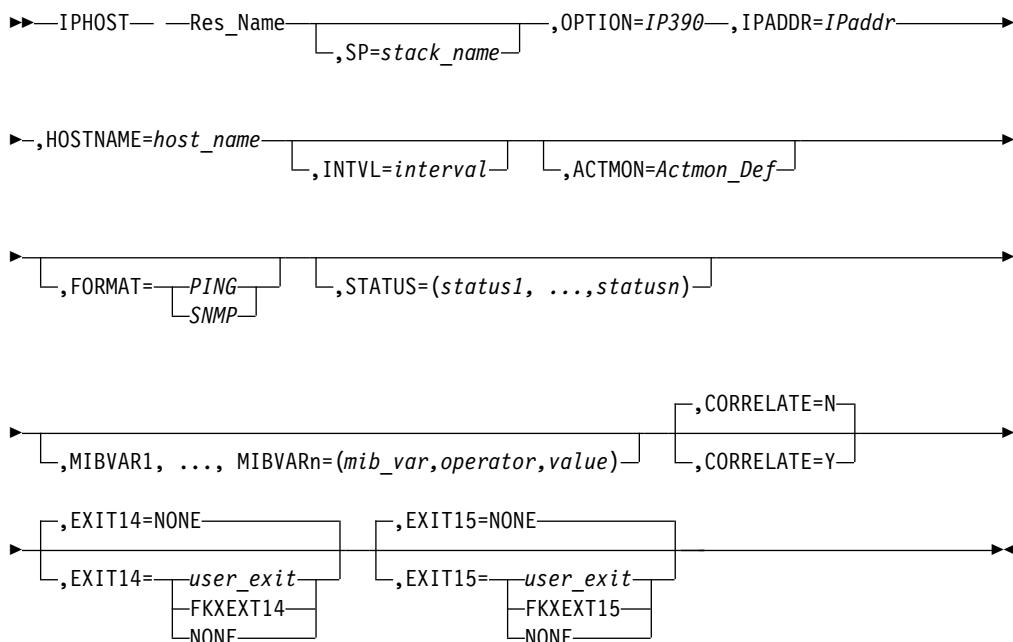
IPHOST

Purpose

The IPHOST control file entry applies to IP management and AON/TCP functions.

The IPHOST control file entry defines an IP Host to IP390 and optionally defines active monitoring of the host.

Syntax



Res_Name

The unique name associated with the TCP/IP host that is used by the NetView program. The name must be a valid REXX symbol.

SP The name of the TCP/IP stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

OPTION=IP390

Only valid entry is IP390.

IPADDR

The IP address of the IP host to be monitored.

This keyword is not required if HOSTNAME is specified.

HOSTNAME

Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The HOSTNAME variable is case-sensitive. The following example shows a host name:

mrpres.whitehouse.capital.usa

This keyword is not required if IPADDR is specified.

INTERVAL

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 409.

FORMAT

Specifies one of the following options to be used to determine the status of a resource.

PING Pings the resource to check its status.

SNMP

Uses SNMP MIB polling to check the status of the resource.

STATUS

Defines the expected status of the resource.

MIBVAR1

Multiple MIBVAR statements can be defined and are needed only for SNMP thresholding (requires FORMAT=SNMP).

You can specify the following values for the *operator*:

- EQ (equal)
- NE (not equal)
- GT (greater than)
- LT (less than)
- GE (greater than or equal)
- LE (less than or equal)

CORRELATE

Set to Y to use trap correlation of IPHOST, IPROUTER, or IPTN3270 and their respective interface. Set the CORRELATE parameter to Y for resources only when SNMP is available. All known interfaces are used to correlate the status. Correlation is only recommended for interfaces installed on critical IPHOSTs or IPROUTERS. N is the default.

EXIT14

Defines additional processing of the SNMP interface table. The default value is NONE. EXIT14 is called for FORMAT=SNMP. The NetView user exit is FKXEXT14.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE. EXIT15 is called for FORMAT=SNMP. The NetView user exit is FKXEXT15.

Usage notes

- The IPHOST statement requires that the IPMGT or AON/TCP tower be enabled.
- If you are passing parameters to a user exit, enclose the value for the keyword in quotation marks.

Examples

The following example defines host myhost for proactive monitoring using SNMP polling. MIB thresholding for myhost is based on MIBVAR1, MIBVAR2, MIBVAR3, and MIBVAR4.

```
ACTMON    IP390,OPTION=IP390,INTVL=01:00,STATUS=NORMAL  
ACTMON    IPHOST,OPTION=IP390,INTVL=00:30
```

```
IPHOST  myhost,SP=NMPIPL10,  
        OPTION=IP390,  
        HOSTNAME=myhost.raleigh.ibm.com,  
        INTVL=45,  
        FORMAT=SNMP,  
        STATUS=(NORMAL,THRESH*,DEGR*),  
        MIBVAR1=(tcpActiveOpens.0,LT,5000),  
        MIBVAR2=(tcpInErrs.0,GT,5),  
        MIBVAR3=(ipInDiscards.0,EQ,1),  
        MIBVAR4=(ipRoutingDiscards.0,GE,3)
```

IPINFC

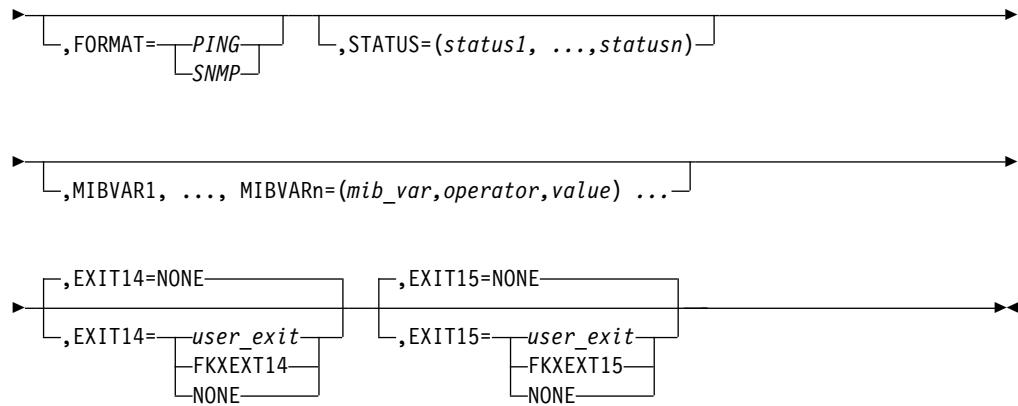
Purpose

The IPINFC control file entry applies to IP management and AON/TCP functions.

The IPINFC control file entry defines an IP interface to IP390 and optionally defines active monitoring of the interface.

Syntax

```
►—IPINFC— —Res_Name— ,OPTION=IP390—,IPADDR=IPaddr—→  
          |,SP=stack_name|  
  
►—,HOSTNAME=host_name— ,INTVL=interval— ,ACTMON=Actmon_Def—→  
          |,INTVL=interval| |,ACTMON=Actmon_Def|
```



Res_Name

The unique name associated with the TCP/IP interface that is used by the NetView program. The name must be a valid REXX symbol.

SP The name of the MVS stack to use for TCP/IP commands. This name is synonymous with the MVS service point.

OPTION

The only valid entry is IP390.

IPADDR

The IP address of the interface to be monitored.

This keyword is not required if HOSTNAME is specified.

HOSTNAME

Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The HOSTNAME variable is case-sensitive. The following example shows a host name:

mrpres.whitehouse.capital.usa

INTERVAL

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 409.

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 are needed only for SNMP thresholding (requires FORMAT=SNMP).

You can specify the following values for the *operator*:

- EQ (equal)
- NE (not equal)

- GT (greater than)
- LT (less than)
- GE (greater than or equal)
- LE (less than or equal)

EXIT14

Defines additional processing of the SNMP interface table. The default value is NONE. EXIT14 is called for FORMAT=SNMP. The NetView user exit is FKXEXT14.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE. EXIT15 is called for FORMAT=SNMP. The NetView user exit is FKXEXT15.

Usage notes

- The IPINFC statement requires that the IPMGT or AON/TCP tower be enabled.
- If you are passing parameters to a user exit, enclose the value for the keyword in quotation marks.

Examples

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=NMPPIPL10,
          IPADDR=9.37.74.18,
          HOSTNAME=mlrite.raleigh.ibm.com

```

IPNAMESERV

Purpose

The IPNAMESERV control file entry applies to IP management and AON/TCP functions.

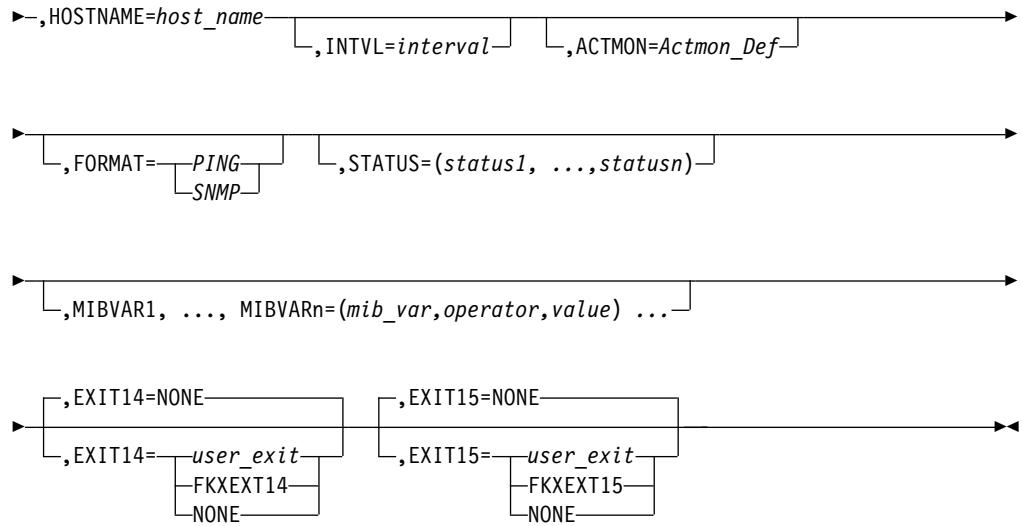
The IPNAMESERV control file entry defines an IP name server to IP390 and optionally defines active monitoring of the NameServer.

Syntax

```

>>—IPNAMESERV— —Res_Name— [ ] —,OPTION=IP390—,IPADDR=IPaddr—→
      ,SP=stack_name

```



Res_Name

The unique name associated with the TCP/IP name server used by the NetView program. The name must be a valid REXX symbol.

SP 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

The IP address of the domain name server to be monitored.

HOSTNAME

Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The HOSTNAME variable is case-sensitive. The following example shows 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 does not occur.

ACTION

Defines active monitoring for resources in the network. For more information, see "ACTION" on page 409.

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 are needed only for SNMP thresholding (requires FORMAT=SNMP).

You can specify the following values for the *operator*:

- EQ (equal)
- NE (not equal)
- GT (greater than)
- LT (less than)
- GE (greater than or equal)
- LE (less than or equal)

EXIT14

Defines additional processing of the SNMP interface table. The default value is NONE. EXIT14 is called for FORMAT=SNMP. The NetView user exit is FKXEXT14.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE. EXIT15 is called for FORMAT=SNMP. The NetView user exit is FKXEXT15.

Usage notes

- The IPNAMESERV statement requires that the IPMGT or AON/TCP tower be enabled.
- If you are passing parameters to a user exit, enclose the value for the keyword in quotation marks.

Examples

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=NMPIPL10,
OPTION=IP390,
IPADDR=9.37.240.3,
HOSTNAME=ralname11.raleigh.ibm.com,
ACTMON=DNSMON
```

IPPORT

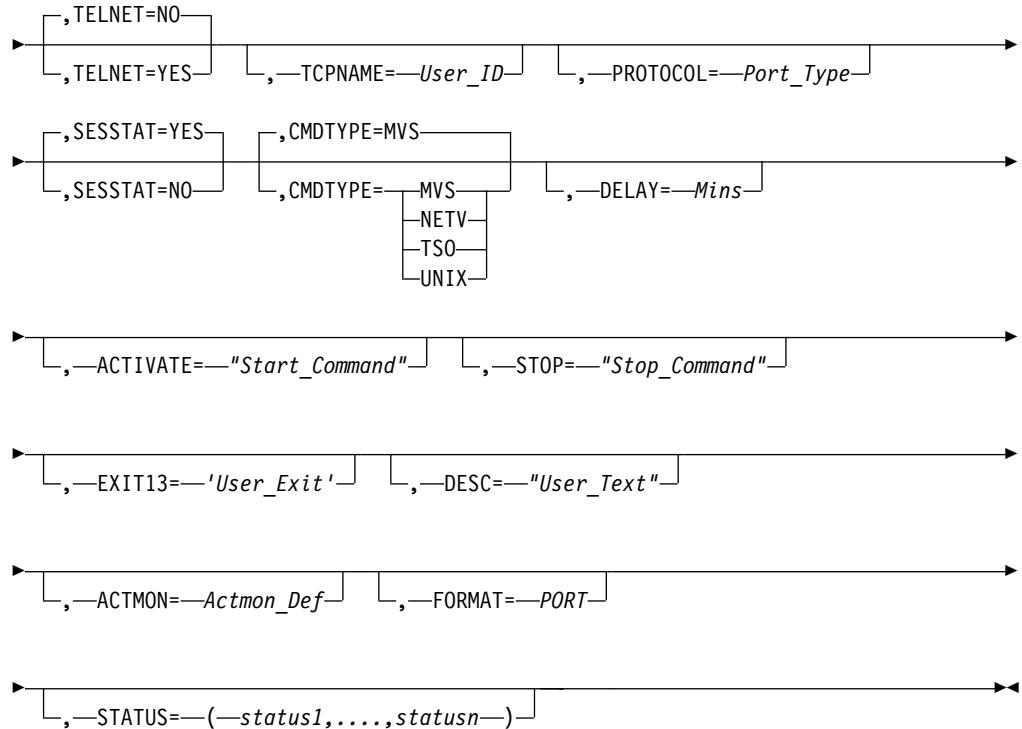
Purpose

The IPPORT control file entry applies to IP management and AON/TCP functions.

The IPPORT control file entry defines IP ports to IP390 and optionally defines monitoring of the specified port.

Syntax

```
>>--IPPORT-- --Socket_Name-- [ ,--SP=--stack_name-- ] [ ,--PORT=--Port_Num-- ] *
```



Socket_Name

The unique name associated with the TCP/IP socket that is used by the NetView program. The name must be a valid REXX symbol.

SP 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 476.

PORT

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 and the resource name is TN3270, the NetView program dynamically determines what TN3270 sockets are known to TCP/IP. You can use this to run multiple TN3270 sockets. Each socket has a policy name of TN- concatenated with the port number. For example, TN-1055 is used for TN3270 sessions connecting into port 1055.

Note: When you are running with multiple TN3270 ports, any changes to the dynamic policy definitions created by the NetView program are not reset until the next stack monitor interval.

TELNET

Indicates whether this port is being used as a Telnet port. The following values are valid:

NO Indicates that this port is not being used as a Telnet port. This is the default if TELNET is not specified.

YES

Indicates that this port is being used as a Telnet port.

TCPNAME

The MVS job name of the application that opened the port.

PROTOCOL

Defines the type of socket. The following types of ports can be specified:

- 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 are displayed when running IPSTAT.

NO When SESSTAT=NO is coded, sessions (if any exist) for that socket are not available to IPSTAT.

CMDTYPE

Defines the type of command. The following types of commands can be specified:

- MVS (default)
- TSO
- UNIX
- NETV

DELAY

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

Defines the command to start the port.

STOP

Defines the command to stop the port.

EXIT 13

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 is not issued.

DESC

The user-defined text.

ACTMON

Defines active monitoring for groups of resources in the network. For additional information see "ACTMON" on page 409.

FORMAT=PORT

Specifies that port data is to be collected to check the status of the resource. PORT is the only valid option for IPPORTs.

STATUS

Defines the expected status of the resource.

Usage notes

The IPPORT statement requires that the IPMGT tower or the AON/TCP tower be enabled.

Examples

- 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.

```
ACTMON    IP390,OPTION=IP390,STATUS=NORMAL  
ACTMON    IPPORT,OPTION=IP390,INTVL=00:05,FORMAT=PORT
```

```
IPPORT SMTP,  
        SP=NMPIPL10,  
        PORT=25,  
        TCPNAME=SMTP32,  
        ACTMON=IPPORT,  
        DESC='SMTP Socket 25'
```

- 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 is also available for the session status functions because SESSTAT=YES, by default.

IPROUTER

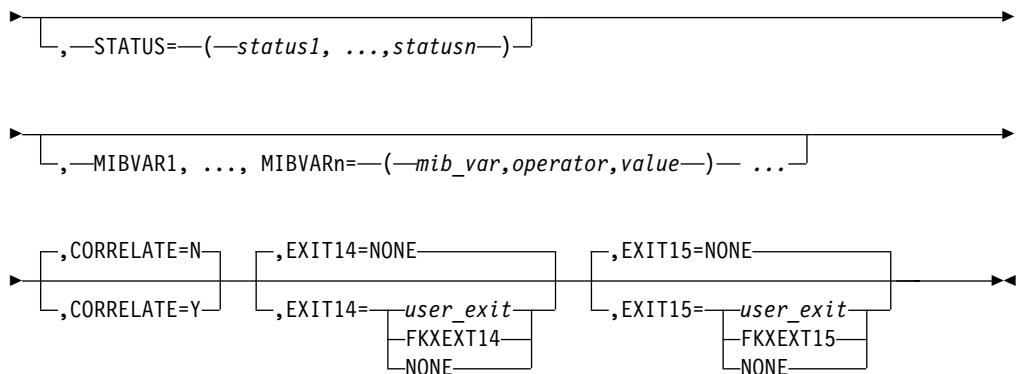
Purpose

The IPROUTER control file entry applies to IP management and AON/TCP functions.

The IPROUTER control file entry defines IP routers to IP390 and optionally defines active monitoring of the router.

Syntax

```
►—IPROUTER— —router_name— [—SP=—stack_name—], —OPTION=—IP390—  
  ►,—IPADDR=—ipaddr—,—HOSTNAME=—host_name— [—INTVL=—interval—]  
  ►,—ACTMON=—Actmon_Def— [—FORMAT=—PING— | —SNMP—]
```



router_name

The unique name associated with the router that is used by the NetView program. The name must be a valid REXX symbol.

SP 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

The IP address of the router to be monitored.

HOSTNAME

Indicates the fully qualified TCP/IP host name, using up to 30 characters including dots. The HOSTNAME variable is case-sensitive. The following example shows 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 does not occur.

ACTMON

Defines active monitoring for groups of resources in the network. For additional information see "ACTMON" on page 409.

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 status of the resource.

MIBVAR1 – n

Multiple MIBVAR statements can be defined and are needed only for SNMP thresholding (requires FORMAT=SNMP).

You can specify the following values for the *operator*:

- EQ (equal)
- NE (not equal)
- GT (greater than)

- LT (less than)
- GE (greater than or equal)
- LE (less than or equal)

CORRELATE

Set to Y to use trap correlation for IPHOSTs, IROUTERS, or IPTN3270s and the respective interface. Set the CORRELATE parameter to Y for resources only when SNMP is available. All known interfaces are used to correlate the status. Use correlation only for interfaces installed on critical IPHOSTs or IROUTERS. N is the default.

EXIT14

Defines additional processing of SNMP interface table. The default value is NONE. Called for FORMAT=SNMP. The NetView user exit is FKXEXT14.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE. EXIT15 is called for FORMAT=SNMP. The NetView user exit is FKXEXT15.

Usage notes

- The IROUTER statement requires that the IPMGT or AON/TCP tower be enabled.
- If you are passing parameters to a user exit, enclose the value for the keyword in quotation marks.

Examples

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    IROUTER,OPTION=IP390,INTVL=00:30,STATUS=NORMAL
ACTMON    NYROUTERS,OPTION=IP390,INTVL=00:15,
          FORMAT=SNMP

IPROUTER  ROUTER1,
          SP=NMPIPL10,
          OPTION=IP390,
          IPADDR=9.67.2.299,
          HOSTNAME=router1.plant.floor.com,
          ACTMON=NYROUTERS
  
```

IPTELNET

Purpose

The IPTELNET control file entry applies to IP management and AON/TCP functions.

The IPTELNET control file entry defines the Telnet server and its defined ports.

Syntax

```
►—IPTELNET— —address_space—,—JOBNAME—=—job_name—,————→
```

```
>--PORT==-(port_number)--,--DOMAIN==NetView_domain_ID-->
```

address_space

The unique name associated with the Telnet address space that is used by the NetView program. The name must be a valid REXX symbol.

JOBNAME

The MVS job name of the Telnet server.

PORT

The port number or numbers that are associated with this Telnet server. An asterisk (*) is not a valid port value.

DOMAIN

The NetView domain associated with this Telnet server.

Usage notes

The IPTELNET statement requires that the IPMGT or AON/TCP tower be enabled.

Examples

```
IPTELNET ADDRSP01,PORT=23,JOBNAME=JOB99,DOMAIN=CNM01
```

IPTN3270

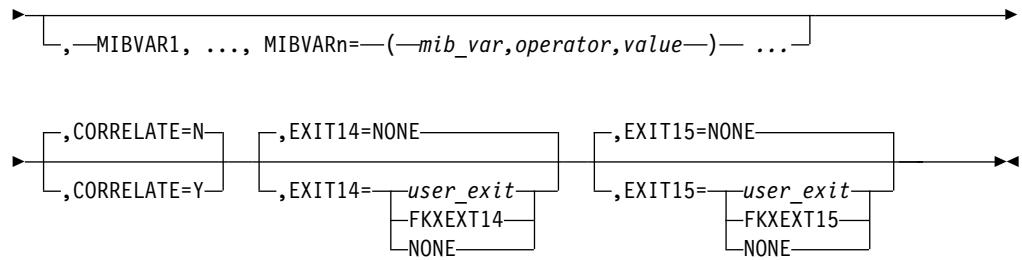
Purpose

The IPTN3270 control file entry applies to IP management and AON/TCP functions.

The IPTN3270 control file entry defines TN3270 servers to IP390 (used for session management) and optionally defines active monitoring for the TN3270 server.

Syntax

```
>>--IPTN3270-- name--,--IPADDR=address--,--SP=stack_name-->  
  
>--,--TNPORT=23--,portnum--,--SYSTEM=TN3270--,--DATACOL=module-->  
  
>--,--DROP=module-- ,--FORMAT=SNMP-- ,--HOSTNAME=host_name-->  
  
>--,--TNIPIADDRx=address:yy-- ,--INTVL=interval-->  
  
>--,--ACTMON=Actmon_Def-- ,--STATUS=(status1, ..., statusn)-->
```



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

The IP address of the TN3270 server to be monitored.

SP 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. The HOSTNAME variable is case-sensitive. The following example shows a host name:

`mrpres.whitehouse.capital.usa`

TNIPADDRx=address:yy

The IP address for the TN3270 server:

x is a single digit integer from 0 - 9

yy is the port number of the Telnet server. The default is 23.

CORRELATE

Set to Y to use trap correlation of IPHOSTs, IPROUTERS, or IPTN3270s and the respective interface. Set the CORRELATE parameter to Y for resources only when SNMP is available. All known interfaces are used to correlate the status. Use correlation only for interfaces installed on critical IPHOSTs or IPROUTERS. 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 409.

STATUS

Defines the expected (AON) status of the resource.

MIBVAR1 – n

Multiple MIBVAR statements can be defined and are needed only for SNMP thresholding (requires FORMAT=SNMP).

You can specify the following values for the *operator*:

- EQ (equal)
- NE (not equal)
- GT (greater than)
- LT (less than)
- GE (greater than or equal)
- LE (less than or equal)

EXIT14

Defines additional processing of the SNMP interface table. The default value is NONE. EXIT14 is called for FORMAT=SNMP. The NetView user exit is FKXEXT14.

EXIT15

Defines additional processing of user-defined thresholds (MIBVAR). The default value is NONE. EXIT15 is called for FORMAT=SNMP. The NetView user exit is FKXEXT15.

Usage notes

- The IPTN3270 statement requires that the IPMGT or AON/TCP tower be enabled.
- If you are passing parameters to a user exit, enclose the value for the keyword in quotation marks.

Examples

- The following example shows IPTN3270:

```
IPTN3270 IBM2210,  
SP=MVSA,  
IPADDR=99.200.99.100,  
TNPORT=23,SYSTEM=TN3270,  
DATACOL=FKXEX216,  
INTVL=15,  
FORMAT=SNMP,  
STATUS=(NORMAL,DEGR*,THRESH*),  
HOSTNAME=IBM2210.rtp.lab.ibm.com
```

- The following example shows IPTN3270 for Cisco CIP.

```
IPTN3270 CIP3270,  
SP=TVT2017,  
IPADDR=99.200.10.100,  
TNIPADDR=99.200.10.101:23,  
TNIPADDR2=99.200.10.102:23,  
TNIPADDR3=99.200.10.103:623,  
SYSTEM=TN3270,  
DATACOL=FKXEXCIP,  
INTVL=15,
```

```
FORMAT=SNMP,  
STATUS=(NORMAL,DEGR*,THRESH*),  
HOSTNAME=cip3270.rtp.lab.ibm.com
```

LSTHRESH

Purpose

The LSTHRESH control file entry applies to base AON functions.

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.

Syntax

LSTHRESH

```
►—LSTHRESH—NAMESERV—,—COUNT—=—(—nn, hh:mm—)—,—EVENTLIM—=—nnn—,—  
►—————  
►—————,NOTIFY=ALL—————  
►—————,NOTIFY=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. Ensure that this number is 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.

Examples

In this example, Name Server alerts are tracked by AON/TCP. A maximum of 50 events are saved in global variables *before wrapping occurs*:

```
LSTHRESH NAMESERV,COUNT=(10,00:10),EVENTLIM=50,NTFY=THR
```

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.

MONIT

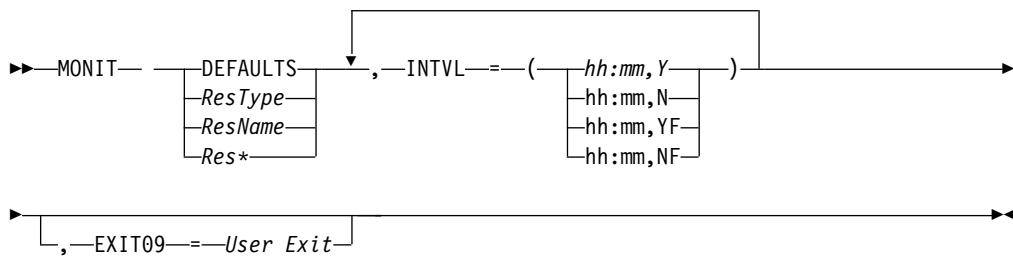
Purpose

The MONIT control file entry applies to IP management and AON functions.

The MONIT control file entries control reactivation intervals for failing network resources. The NetView program 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.

Syntax

MONIT



DEFAULTS

System wide defaults. A MONIT DEFAULTS entry is required.

ResType

Type of resource to automate. The NetView program recognizes the following resource types:

- Subsystems
 - MVS components
 - 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 wild card
 - % Single character wild card

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 actions to take at each interval. No default value exists.

- Y Calls the Notification Policy defined for the resource or event type. Recovery monitoring ends after the last interval that is specified.
- N Does not call any Notification Policy. Sends messages to the notification operators. Recovery monitoring ends after the last interval that is specified.
- YF Notify operators based on your NOMOMONS Notification Policy. Recovery monitoring continues until the resource recovers or the control file is reloaded, using the last interval that is specified.
- NF Repeat recovery monitoring at the last interval that is specified until the resource recovers or the control file is reloaded. Do not call the 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 notes

- The MONIT statement requires that the IPMGT or AON tower be enabled.
- The NetView program 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, the NetView program uses the MONIT statement for the resource type PU. If PU02 fails, the NetView program 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 TA1T100 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

Purpose

The MONITOR control file entry applies to base AON functions.

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.

Syntax

```
>>>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 re-initializing the logon process.

Usage notes

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 reestablish the NNT session.

Examples

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
```

NCPRECOV (SNA)

Purpose

The NCPRECOV control file entry applies to base AON functions.

The NCPRECOV control file entries in the control file specify recovery parameters for NCPs. When an option to dump the NCP is received, AON/SNA responds with noncritical replies from the control file. Critical replies are sent when the NCP dumps enough times to cause the critical threshold setting for the NCP is to be exceeded. An alert is sent to the notification operator when the dump reply is made. The notification operator is also alerted if the dump fails or takes longer than the time defined in the DUMPTIME parameter.

AON/SNA uses the NCPRECOV values to determine whether to reload the NCP. As with the dump option, noncritical replies are issued until the thresholds for the number of allowed replies are exceeded. An alert is issued upon reply or failure of the critical load. The critical threshold for the number of dumps within a certain time period for an NCP is defined in the THRESHOLDS control file entry. CP activation also generates an alert.

If the initial test routine cannot be accessed, AON/SNA bypasses the initial test and loads the NCP. The notification operator is alerted if a load failure or a session failure occurs between the NCP and systems services control point (SSCP).

When the NCP is loaded with the wrong load module, AON/SNA cancels activation of the NCP and alerts the notification operators.

NCPRECOV is an optional control file entry. However, you must define an NCPRECOV statement for each NCP to be recovered on the owning host and any channel attached host running AON/NCPRECOV.

Syntax

```
►—NCPRECOV— —ncpname—,—HOST—=—domain—  
  ,—DUMP—=(—Y—, —Y—)—, —RELOAD—=(—Y—, —Y—)—  
  ,—DUMPTIME—=—hh:mm—, —LOADTIME—=—hh:mm—  
  ,—LINKSTA—=—resource_name—  
  ,—LOADMOD—=—loadmod_name—  
  ,—DUMPSTA—=—resource_name—  ,—EXIT01—=—cmd—  
  ,—EXIT02—=—cmd—  ,—AUTOINIT—=—Y—  ,—EXIT03—=—cmd—  
  ,—EXIT04—=—cmd—  
►
```

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*) required by a dump of this NCP, after which an operator is notified that excessive time has been used. 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*) required by a load of this NCP, after which an operator is notified that excessive time has been used. 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 link station 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 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 is 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 notes

- 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 can code AUTOINIT=N for any less critical NCP if you have a large network.

Examples

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.

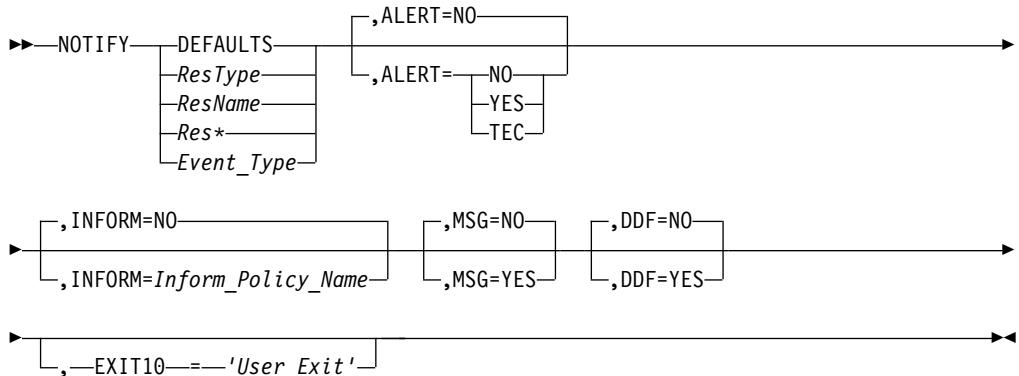
NOTIFY

Purpose

The NOTIFY control file entry applies to IP management and AON functions.

The NOTIFY control file entry defines notification actions for the NetView operators who receive notifications. These notifications can be messages, alerts, DDF updates, and Inform Policy actions; such as, beeper or email.

Syntax



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 have one of the following values:

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 Event Integration Facility (EIF) 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 designated event server receives the notification.

INFORM

Specifies the use of pager/beeper or email that is defined in the CONTACT entry 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.

Usage notes

The NOTIFY statement requires that the IPMGT or AON tower be enabled.

Examples

- This example defines DEFAULTS Notification Policy as Event Integration Facility (EIF) updates, and DDF updates for all resources. Messages are not passed:

```
NOTIFY DEFAULTS,ALERT=TEC,MSG=NO,DDF=YES
```

EIF updates occur and DDF records are created.

- This example defines Notification Policy for an event type of CRITTHRS, Critical Threshold Exceeded. In this case, an INFORM Policy of Critical is called to email support personnel. If MSG=YES is coded then messages are generated.

```
NOTIFY CRITTHRS,  
        INFORM=CRITICAL,  
        MSG=YES
```

NTFYOP**Purpose**

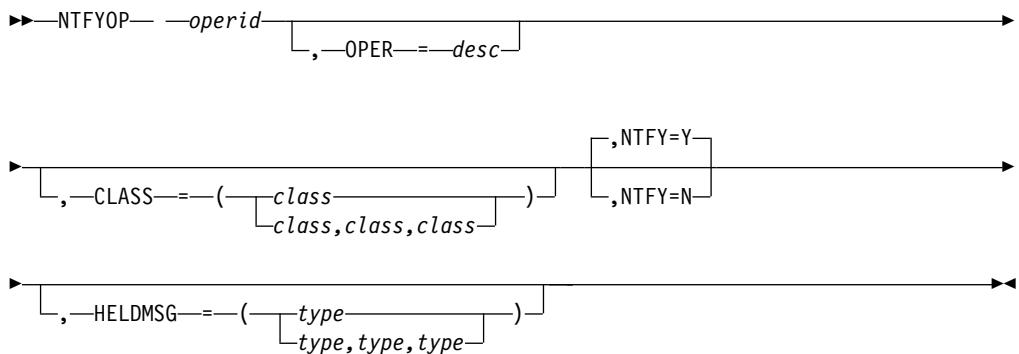
The NTFYOP control file entry applies to IP management and AON functions.

The NTFYOP control file entry identifies NetView operators who receive notifications. These notifications are messages sent to the operator workstation and can be based on job responsibilities, such as all IP name servers. The messages can be held until the operator clears them. Assign operators responsible for system operation to receive all messages or a subset of those messages.

NTFYOP is a required control file entry. There must always be one NTFYOP logged on to ensure all messages are eligible for automation. Consider using an AUTOOPER for automation.

Syntax

NTFYOP



operid

Identifies the NetView operators to receive notification messages. This name cannot exceed 8 characters, and is defined in the member DSIOPF. 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 can be up to 20 characters long and must be enclosed in quotation marks if it contains blanks.

CLASS

Specifies the message notification classes to be sent to this operator ID. Up to 10 message classes can be specified. No default message classes exist. The class assigned to the notify operators can be changed using the SETNTFY command. The following table lists the AON message classes:

Table 16. AON Message Classes

Message Class	Message Text
00	Errors
01	Down
02	Up
03	Degraded/Performance
04	Manual Intervention Required
07	Critical Threshold Exceeded
08	Frequent Threshold Exceeded

Table 16. AON Message Classes (continued)

Message Class	Message Text
09	Infrequent Threshold Exceeded
10	SNA Automation
11	VTAM Detected Storage Problems
12	X.25 Automation
20	VTAM Subarea Automation
21	NCP, Linksta
22	Line
23	PU
24	CDRM
25	CDRSC
26	APPL
27	Session
40	Messages to forward to System Automation for z/OS
50	Advanced Peer-to-Peer Networking Automation
60	TCP/IP Automation
61	IP Router
62	Nameserver
63	IP Socket
64	MVS TCP/IP Automation
65	Tivoli NetView for UNIX automation
66	Service Point
67	IP Host
68	Interface
69	IP Link
90	AON Automation

NTFY

Specifies whether the operator receives notification messages. The default is YES.

HELDMSG

Specifies the type or types of messages that are held on the operator screen. The first character of the HELDMSG type specifies the message type. Multiple types can be specified. The following message types can be specified:

I or INFO

Informational messages

W or WARN

Warning messages

E or ERROR

Error messages

A or ACTION

Action messages

Usage notes

- The NTFYOP statement requires that the IPMGT or AON tower be enabled.
- The NTFY=Y|N can be used to temporarily turn on and off receiving notification messages. The NTFY parameter can be changed while the NetView program 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 notification messages are received and a NTFYOP operator is not logged on to the NetView program, messages are sent to the log and passed through automation. However, the messages do not trigger automation. A NTFYOP must be logged on for the messages to be eligible for automation.
- 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.

Examples

- The following example defines an operator to receive AON notifications only:
`NTFYOP NETOP1,OPER='NETOPER 1',CLASS=(10)`
NETOP1 is defined to receive AON notifications. The notify parameter was not coded and defaults to NTFY=Y.
- The following 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`

SYSOP is defined to receive notifications 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.

RECOVERY

Purpose

The RECOVERY control file entry applies to IP management and AON functions.

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.

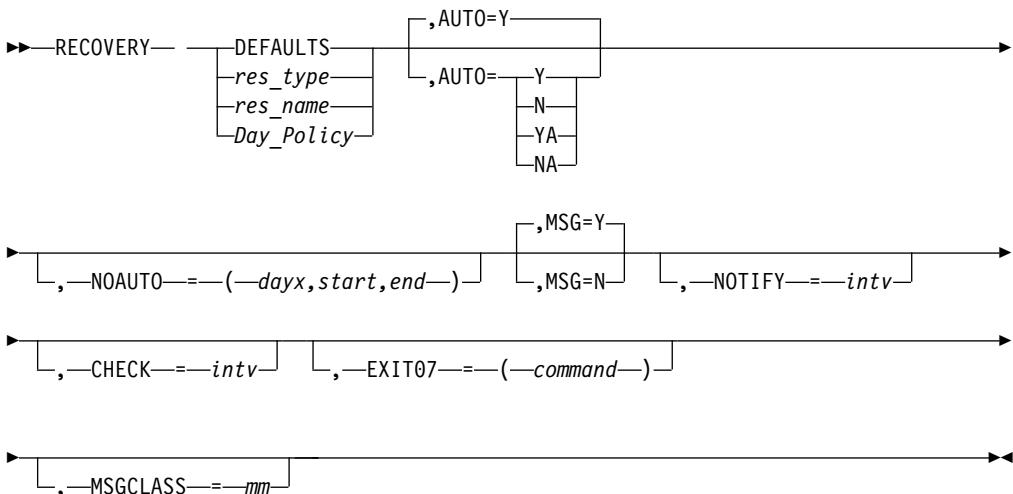
The RECOVERY entry is required. A RECOVERY DEFAULTS entry must be specified.

The NetView program is designed to meet most recovery processing needs. However, your installation might have special requirements that require you to tailor recovery automation processing. You can code user exits for specific recovery processing. User exits for recovery processing are specified on the RECOVERY statement or on the ENVIRON EXIT statement.

RECOVERY monitoring is invoked by message automation and by active monitoring when it is determined that a resource is not active. The processing completes when the NetView program determines that the resource is recovered. Before completion, recovery monitoring checks for ACTMON definitions and resumes active monitoring when appropriate.

Syntax

RECOVERY



DEFUALTS

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_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 wild card
- % Single character wild card

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.

res_name

The name of the resource to automate.

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
WEEKDAY
WEEKEND
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 because
only 3 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.

- NA** 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 processing can cause a significant amount of overhead.

During RECOVERY processing, the NetView program checks for the presence of any special day specifications in the NetView CHRON command calendar

schedule file, DSISCHED. The RECOVERY process compares any keywords returned from DSISCHED to your RECOVERY control file entries to determine whether recovery processing takes place.

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
- WEEKDAY
- WEEKEND
- 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 because 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 - 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 - 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, EZL509I, and EZL510I) for failures and recoveries are sent. This includes updating DDF and log files.

Note:

1. If AUTO=N and MSG=N, no messages about this resource are included in AON logs or DDF.
2. If AUTO=Y and MSG=N, operators continue to receive messages concerning recovery monitoring, active monitoring, and thresholding.

NOTIFY

Defines the interval that the NetView program uses to notify the operators if the APPLICATION is not active. This option is only used for APPL resource types.

intv

Defines the time interval that the NetView program uses to notify the

operators that the application is still not operational. You must specify the time using the 24-hour clock in the *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.

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 *hh:mm* format. Values can be in the range 00:00 - 23:59.

EXIT0

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 notes

- The RECOVERY statement requires that the IPMGT or AON tower be enabled.
- 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. DEFAULTSThe DEFAULTS NCP entry is required.
- A RECOVERY DEFAULTS entry is required if a RECOVERY NCP statement is defined.
- 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.

Examples

- The following example defines a sample DEFAULTS entry. In this example, recovery activity for all resources are automated.

```
RECOVERY DEFAULTS,AUTO=Y
```
- 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)
```
- The following example defines an entry where automation is off Sunday evening through Monday morning. In this example, RECOVERY is off from 6 p.m. on Sunday until 2 A.M. 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)
```

- 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
```
- 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
```
- 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 and an SNA application is specified.

```
RECOVERY APPL,AUTO=N
RECOVERY CICS1,AUTO=Y,NOTIFY=01:00,CHECK=00:15
RECOVERY MSM*,AUTO=Y
```
- 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 is 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 and noon on the last Saturday of every month.

```
RECOVERY ABC*,AUTO=Y,NOAUTO=(*/SAT/LAST/*,00:00,12:00)
```

RESIDENT

Purpose

The RESIDENT control file entry applies to base AON functions.

The RESIDENT control file entries in the control file define programs that are loaded into main storage to improve performance.

Syntax

►►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.

Examples

The following example defines the program NEWONE, which AON loads during initialization.

```
RESIDENT NEWONE
```

SESSION

Purpose

The SESSION control file entry applies to base AON functions.

The SESSION control file entry is used to identify sessions that are to be actively monitored by AON/SNA.

Syntax

SESSION

```
►►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.luname*.

Usage notes

- The Advanced Peer-to-Peer Networking architecture 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=RALSND$1,LU2=RALSND$8  
SESSION RJE2,LU1=RALSND$1,LU2=RALSND$7  
SESSION RJE3,LU1=RALSND$1,LU2=RALSND$6
```

SUBSYSTEM (SNA)

Purpose

The SUBSYSTEM control file entry applies to base AON functions.

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.

Syntax

```
►►—SUBSYSTEM— —subsystemname—,—JOB—=—appname————→  
►,—DESC—=—'netview_access'————→
```

subsystemname

Name of the DB2 subsystem to which NetView connects. *Subsystemname* is a character value with a length of 1 - 4 characters.

For example, **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 notes

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.

Examples

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

Purpose

The TCP390 statement applies to base NetView policy statements.

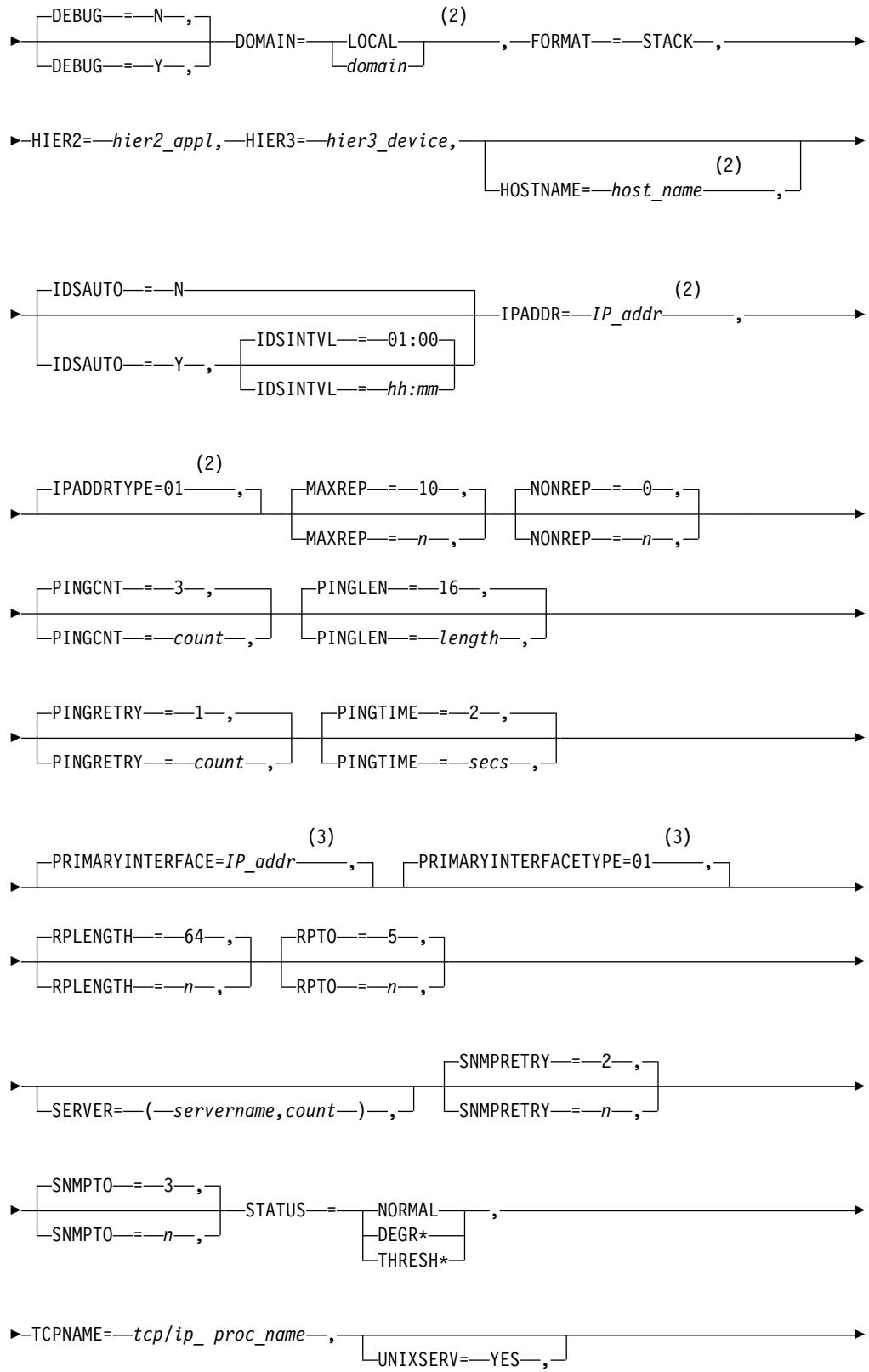
The TCP390 policy statement defines a TCP/IP stack to the NetView program. A TCP390 entry is required for each stack that you want to manage.

Syntax

The TCP390 policy statement has the following syntax:

TCP390

```
►►—TCP390— —tcpip_stack————(1)————, —COMMUNITYNAME—=—public—,————→  
|————DEFUALTS————|————, —COMMUNITYNAME—=—community_name—,————|
```



► [VERBOSER— -0—value] ►

Notes:

- 1 This must be a short name.
- 2 This parameter does not need to be specified when resource discovery is enabled.
- 3 Do not specify this parameter. It is set by resource discovery.

where:

tcpip_stack

Specifies the host name of the TCP/IP for MVS stack. Use the TCP/IP short name. This name must consist of characters supported by the GLOBALV command. For example, MY_STACK is acceptable, but MY-STACK is not.

DEFAULTS

Specifies the keywords and values that are used for TCP390 statements when a keyword is not explicitly specified on the statement. You can include the following keywords on the TCP390.DEFAULTS statement:

- DEBUG
- MAXREP
- NONREP
- PINGCNT
- PINGLEN
- PINGRETRY
- PINGTIME
- RPLENGTH
- RPTO
- SNMPPRETRY
- SNMPTO
- VERBOSE

COMMUNITYNAME

Defines the SNMP community name to be used for all SNMP requests that apply to the stack. The default value for the community name is **public**. 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 causes errors. For more information about defining the community name to TCP/IP, see the *z/OS Communications Server IP Configuration Reference*.

DEBUG=N | Y

Only applies to AON/TCP option 2.5 (Issue SNMP Commands). This determines whether additional diagnostic information is provided when issuing SNMP commands.

N Indicates that no additional diagnostic information is provided. This is the default.

Y Indicates that additional diagnostic information is provided.

DOMAIN=LOCAL | domain

Specifies the NetView domain that owns this MVS service point:

LOCAL

Indicates the same host as this NetView program.

domain

The remote NetView domain ID.

FORMAT=STACK

Defines the 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_app1

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.

IDSAUTO=N|Y

Defines whether the stack supports Intrusion Detection Services (IDS) automation.

N No indicates that the stack does not support IDS automation.

Y Yes indicates that the stack supports IDS automation.

This parameter can only be used for a local stack. If the IDS subtower is enabled, then by default, the IDSAUTO parameter is set to Y for the local stacks. You can override this default setting by adding the IDSAUTO=N parameter to the TCP390 definition for each stack.

IDSINTVL= hh:mm

Specifies the time interval during which to schedule the IDS report generation timer.

hh Specifies the hour in the range of 0 - 24.

mm Specifies the minutes in the range of 0 - 59.

Note:

1. This parameter is ignored if IDSAUTO=N.
2. If you do not specify IDSINTVL but specify IDSAUTO=Y, probe summary statistics reports are not generated.

IPADDR=IP_addr

Specifies the IP address for this stack. The *IP_addr* is required. You can also specify *hostname* or both. If *IP_addr* is not specified, the NetView program attempts to determine the address and adds it to the in-storage copy of the policy definitions.

IPADDRTYPE

01 indicates the address is using IP Version 4 (IPv4) format.

02 indicates the address is using IP Version 6 (IPv6) format.

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.

PINGCNT= 3 | count

Specifies the number of times to issue the PING. Valid values are in the range of 1 - 999. The default is 3.

PINGLEN= 16 | length

PINGLEN is the number of bytes in the echo request. Valid values are in the range of 8 - 65487. The default is 16.

PINGRETRY= 1 | count

Specifies the number of PINGS to issue during ACTMON processing.

PINGTIME= 2 | secs

Specifies the length of time to wait before timing out. Valid values are in the range of 1 - 100. The default is 2.

PRIMARYINTERFACE=IP_addr

Specifies the address of the primary interface on the stack.

Note: This parameter is set by resource discovery.

PRIMARYINTERFACETYPE

01 indicates the primary interface is using the IPv4 format.

Note: This parameter is set by resource discovery.

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 is to 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.

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* is appended to *Servername* to form a valid TSO SERVERNAME.

SNMPRETRY= 2 | n (SNMP Option)

Specifies the number of times the SNMP command is to 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 is to wait for a response. Valid values are in the range of 1 - 30. The default is 3.

STATUS= status

Specifies the status.

NORMAL

Indicates that the status is normal.

DEGR*

Indicates that the status is degraded.

THRESH*

Indicates that the threshold is met.

Note:

1. This parameter is ignored if IDSAUTO=N.
2. If you do not specify IDSINTVL but specify IDSAUTO=Y, probe summary statistics reports are not generated.

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.

UNIXSERV=YES

UNIXSERV defines this stack as a UNIX environment. This parameter must be defined for all service points using UNIX to communicate, including local and remote.

VERBOSE=-*Ovalue*

Only applies to AON/TCP option 2.5 (Issue SNMP Commands). Enables a user to specify output options for the SNMP commands. See the SNMP command help for the -O output option syntax and values.

Usage notes

- The TCP390 statements are located in member CNMPOLCY.
- Resource discovery dynamically determines your local and remote stacks and automatically creates the corresponding TCP390 policy definitions.
- If you define TCP390 statements, any parameters that you specify override dynamically discovered data.
For example, you might choose to code a TCP390 policy definition to force automation to use a specified IPADDR or to run IDS automation services.
- Some TCP390 keywords relate only to AON/TCP automation. If you are not running AON/TCP, those keywords are not in the policy definition after resource discovery completes.
- Each stack is monitored with its own timer (FKXEACT2).

TGSWITCH (SNA)

Purpose

The TGWSWITCH control file entry applies to base AON functions.

The TGWSWITCH 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, the TGWSWITCH is used to return to the primary line by inactivating the backup line when the primary line becomes active. When the two lines are of unequal speeds, ensure that the high-speed line is the primary data carrier.

Syntax

TGSWITCH

```
►►TGSWITCH— —prim_line—,—BKUP—=—bkup_line—————►►
```

prim_line

The primary line, usually the higher speed

BKUP

The backup line, usually the lower speed

Usage notes

Ensure that both lines are part of the same VTAM transmission group (TG).

Examples

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 goes by way of TA1L1004.

```
TGSWITCH TA1L1004,BKUP=TA1L1005
```

THRESHOLDS

Purpose

The THRESHOLDS control file entry applies to base AON functions.

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. Operators can then 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 can 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 might 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.

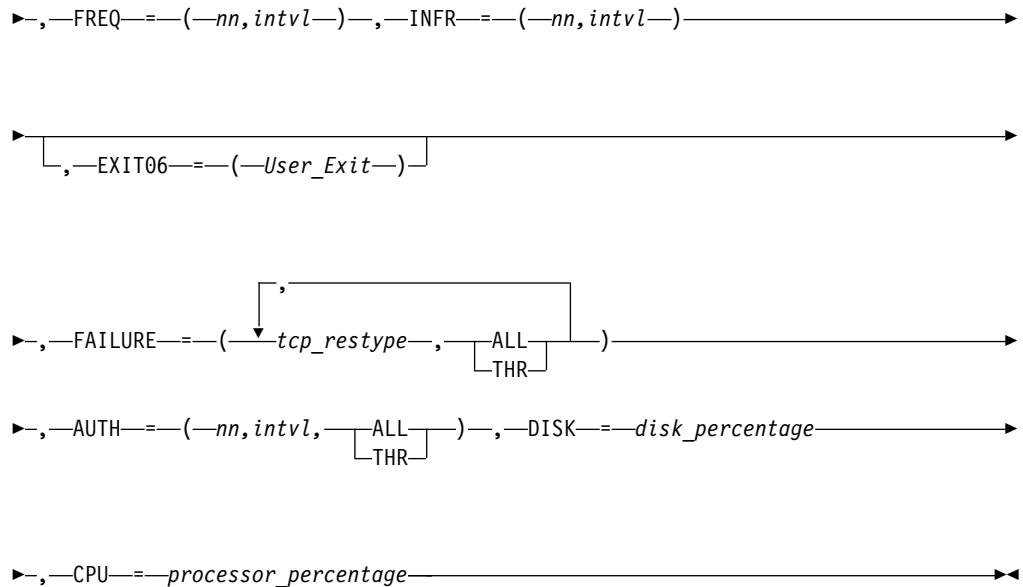
Syntax

```
►►THRESHOLDS— —DEFUALTS—,—CRIT—=—(—nn,intvl,—

|   |
|---|
| Y |
| N |

—)—————►►
```

ResType
ResName
*Res**



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.

ResType

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 wild card

% Single character wild card

For example, both PU0* and P%1 match for PU01.

ResName

The name of the resource to be automated. This resource name can be a network resource type, resource name, or resource name mask (for the naming conventions for an installation). Examples of resources names are PU01 and LINE23. This is the most specific type of THRESHOLDS entry.

*Res**

The group of resources levels to be automated

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 (1 - 10) within the time period that is specified in *intvl* that define the critical threshold.

intvl Time period in hours and minutes (*hh:mm*) for the critical threshold occurrences specified in *nn* in the range of 00:01 - 99:59.

Y|N Specifies whether the critical threshold exception calls the CRITTHRS Notification Policy. The default value is Y.

FREQ

Specifies the frequent threshold occurrences and time interval.

nn Number of occurrences (1 - 10) within the time period specified in *intvl* that define the frequent threshold.

intvl Time period in hours and minutes (*hh:mm*) for the frequent threshold occurrences specified in *nn* in the range of 00:01 - 99:59.

INFR

Specifies the infrequent threshold occurrences and time interval.

nn Number of occurrences (1 - 10) within the time period specified in *intvl* that define the infrequent threshold.

intvl Time period in hours and minutes (*hh:mm*) for the infrequent threshold occurrences specified in *nn* 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 values:

tcp_restype

Indicates the resource types for which thresholding analysis takes place. More 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).

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 values:

nn 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. This value can be in the range of 1 - 99. When the security failure threshold is met, operators are notified of the condition.

intvl

Defines the interval of time, in hours and minutes (*hh:mm*), in which AON/TCP is counting security authorization failures for thresholding. The *intvl* value is specified in the range of 00:01 - 99:59.

THR|ALL

Defines whether the operator is to be notified every time a security authorization failure (ALL) occurs 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 processor utilization percentage that exceeds the amount defined in *processor_percentage*. AON/TCP notifies operators of a processor utilization threshold exception. This value can be in the range of 0 - 99.

Usage notes

- Ensure that the THRESHOLDS entries 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.

Examples

- In this example, the following system defaults are defined:

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)

- In the following example, PU thresholds are defined in the following way:

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

Purpose

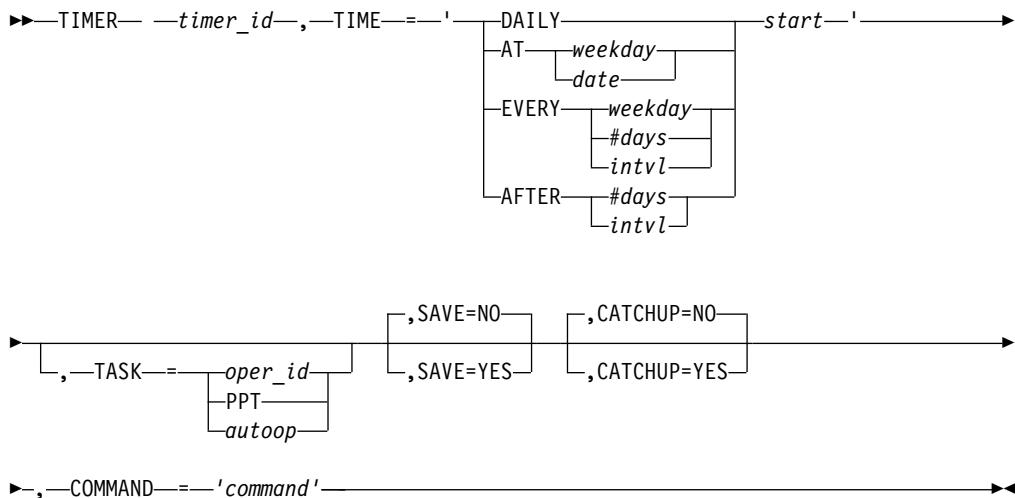
The TIMER control file entry applies to base NetView policy statements.

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.

Syntax

TIMER



timer_id

The identifier that you specify for this timer request. The *timer_id* can be from 1 - 8 characters in length. You cannot use the following values for *timer_id*:

- The word ALL
- The characters EZL, FKV, FKW, FKX, SYS, or RST as the first three characters
- Any VTAM resource name or Dncpname or Rncpname.

Note: If you use any of these values, automation can be unpredictable.

TIME

Specifies when or how often you want the command issued.

DAILY

The NetView program issues the command daily.

- AT** The NetView program issues the command at the specified time. If the date defaults and the time specified is earlier than the current time, the command is scheduled for the next day.

EVERY

The NetView program issues the command repetitively at a timed interval. The EVERY parameter is not valid with a date.

AFTER

The NetView program 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*, the NetView program issues the timer command only if the operator is logged on.

PPT

Specifies that the command or command list indicated by the COMMAND parameter runs 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. For example, the following commands cannot run under the PPT:

- Commands that control the screen (such as AUTOWRAP, INPUT, and SET PF*nn*).
- 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 415.

SAVE

Indicates whether this timer event is 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 was 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 start time for the timer, 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 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 can 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 the NetView program was installed can 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.
- To avoid overloading system resources, do not schedule an excessive number of commands during short time intervals.
- If you schedule commands under the PPT, they might not run in the order that you specify if the value of the time parameter is the same for each command.

Examples

- In this example, the CDRMS command list is issued on August 28, 2009, at 5:20 P.M. The timer is initiated only if operator NETOP2 is logged on. If a system failure occurs prior to 17:20 and restarts after 17:20, this timer is initiated because CATCHUP=YES is specified.

```
TIMER RUNCLST,TIME='AT 08/28/09 17:20',SAVE=YES,TASK=NETOP2,  
CATCHUP=YES,COMMAND='CDRMS'
```
- In this example, the command is issued at 5:21 P.M. on Thursday to load NCP21. If a system outage occurs, this timer does not catch up 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'
```
- The D NET,BFRUSE command is issued at noon on every Sunday. The operator BJSIMPS must be logged on. This timer does not catch up because the EVERY parameter was specified.

```
TIMER DISPLAY1,TIME='EVERY SUNDAY 12:00',TASK=BJSIMPS,  
CATCHUP=YES,COMMAND='D NET,BFRUSE'
```
- In this example, the D NET,CDRMS command is issued after 20 hours and 40 minutes. Because no task keyword is 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'
```
- In this example, the MVS D J,L command is issued 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'

```
- In this example, the specified command is issued at 6:15 P.M. on August 2, 2009, 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 8/02/09 18:15',SAVE=YES,TASK=PPT,
              CATCHUP=YES,COMMAND='D NET,ID=CICSAPPL'

```
- In this example, after 6 minutes, the VARY NET,ACT,ID=CDRMAB command is issued.
- ```

 TIMER DISPLAY4,TIME='AFTER 6',SAVE=YES,
 COMMAND='VARY NET,ACT,ID=CDRMAB'

```
- 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'

```
- In this example, the MVS S SYSLOG command is issued every 20 minutes. This timer runs on the operator ID performing Timer initialization. Usually, AUTO1 performs initialization during NetView startup, therefore, this timer is scheduled to initiate under AUTO1 if this entry is defined in the control file.
- ```

 TIMER MVSSTRT,TIME='EVERY 20',SAVE=YES,
 COMMAND='MVS S SYSLOG'

```

## TSOSERV

### Purpose

The TSOSERV control file entry applies to base AON and IP management functions.

Defines the name of the MVS procedure to use when starting the TSO servers.

### Syntax

```

 ▶▶—TSOSERV— —server— [,PROC=CNMSJTSO] [,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 *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components*.

## X25MONIT (SNA)

### Purpose

The X25MONIT control file entry applies to base AON functions.

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.

### Syntax

#### X25MONIT LINE

```
►--X25MONIT LINE--n --,--X25MCH==linename --,--MCHGRP==groupname ----->
 |,--LOLCN==lowest_lcn --,--HILCN==highest_lcn ----->
 |,--CALLTYPE==IN | OUT |,--TIMER==min |,--THRESH==count ----->
 |,--INOUT----->
```

*n* Specifies a sequential line count, 1 to *n*.

#### X25MCH

Specifies the *linename* on the X25MCH macro.

#### MCHGRP

Defines the NCP *groupname*, which contains the X.25 definitions.

#### LOLCN

Defines the *lowest\_lcn*, which is the lowest logical channel number (LCN) in the group.

#### HILCN

Defines the *highest\_lcn*, which is the highest LCN (logical channel number) in the group.

#### CALLTYPE

Defines the characteristics of a logical group channel. The parameter choices are IN, OUT, and INOUT.

#### TIMER

Sets a timer to monitor switch virtual circuits (SVCs) status at the specified number of minutes (1–99 minutes).

#### THRESH

Sets the minimum number of free switched virtual circuit (SVCs) for which the user wants to be notified. The number must be between the number set in the HILCN and LOLCN parameters.

### Usage notes

- 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'gnn'* and *nn=lcn*.

where

*g* The logical group channel number.

*nn* The lowest logical channel number operand.

- Define the X25MCH and MCHGRP parameters with the following NPSI naming convention:

- X25MCH must be *pLxxxlcn*

- MCHGRP must be *p25Sxxxxy*

where

*p* The prefix coded on the X25BUILD macro.

*xxx*

The address on the ADDRESS operand of the X25MCH macro.

*lcn*

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.

## Examples

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) is 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 5. Inform Policy Member

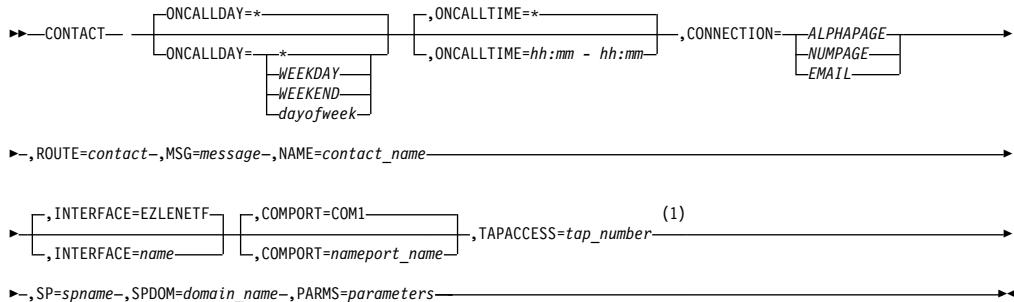
The INFORM policy member is used to define which personnel are to be contacted, when to contact them, and how to contact them. By default the INFORM policy provides support for email, numeric and alphanumeric pagers. Using the SETUP, GROUP, INFORM, and CONTACT statements that follow defines your INFORM policy member. See the INFORMTB command in the *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* for more information. Refer to the *IBM Tivoli NetView for z/OS User's Guide: Automated Operations Network* for more information about 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. Only use these statements before, between, and following completed inform member statements. Using %INCLUDE statements from within inform member statements can produce unpredictable syntax error messages, if the %INCLUDE statements fail.
- For interfaces other than EZLENETF it might be necessary to place dummy values in the inform policy member for some keywords (for example, SP=).
- SMTP can be used for email 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. Because of the I/O required, inform logging is not recommended in all installations.

### CONTACT



### 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.

*hh:mm*

Specifies the on call stop time, where *hh* is the hour in the range of 00 - 24, and *mm* is the minutes in the range of 00 - 59. The stop time must be later than the start time.

**Note:** 24:00 hours represents midnight and is supported for readability of the policy.

**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.

**ALPHAPAGE**

Alphanumeric pager

**NUMPAGE**

Numeric pager

**EMAIL**

Electronic mail

**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 email address. A maximum of 80 characters is supported.

*phone\_#*

A string of integers containing all access numbers, area code, and phone number.

*email\_address*

A continuous string of characters that includes the user ID and address. For example, SantaClaus@NORTH.POLE.COM.

**MSG**

Specifies the messages that are to be sent. The message type and format are dependent upon the CONNECTION type selected.

For CONNECTION=ALPHAPAGE or 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 the following synonym replacements for calls made through EZLENFRM:

**%mm%**

Month

**%dd%** Day

**%yy%** 2-digit year

**%yyyy%**

4-digit year

**%hr%** Hour

**%mn%**

Minute

**%policyname%**

Replaced with the name of the current policy

**%resname%**

Replaced with the resname

**%restype%**

Replaced with the restype

**%resdomain%**

Domain of the failing resource

**%resstat%**

Replaced with the resources status at time of failure

**%aostat%**

Replaced with the automation status

- + 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 is combined with no space between the data on the two lines.

**default**

The following message is a default message:

INFORM FOR %restype% %resname% STATUS=%resstat% DOMAIN=%resdomain%

**Note:** Message length must be handled by the INTERFACE.

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 in the following way: 1-(919) 123-4567. The following characters are 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 callback 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, EZLESMTF 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 in the following way: 1-(919) 123-4567. The following characters are 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 is 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 might 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 are 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 uses the inform policy for PersonC. PersonC is available every evening by email, 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=IBPERSC@VNET.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:

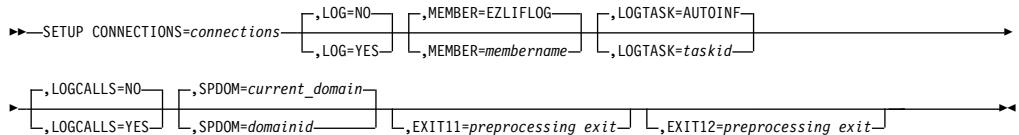
The contact entry supports mixed case.

---

## 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 run time. 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 EZLENFRM 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. Only log NOTIFY actions. As needed, operator INFORM actions can also be logged by setting LOGCALLS=Yes.

#### **SPDOM**

The domain ID of the network NetView program 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 called before 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 following inputs are from EZLENFRM:

```
policy name,resource name,resource type, domain id,
resource status, automation status id
```

The following inputs are from EZLECALL:

```
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. Before calling the interface specified in the policy, and passing the entire policy which includes the error or operator message, Exit 12 is called. 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 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 IBM Tivoli NetView for z/OS. Where applicable, compile exits and load them into storage.

#### **Example:**

This example adds a setup statement to define other connection types and enables logging. It specifies the log member name as INFLOG (the default is EZLIFLOG). A customer written interface is required to implement the FAX connection.

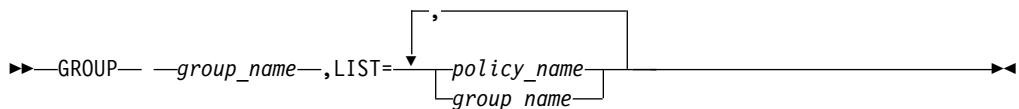
```
SETUP CONNECTIONS=FAX,LOG=YES, MEMBER=INFLOG;
```

---

## GROUP

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 might be contacted. Using these definitions, each person can then be added to any number of different groups. This eliminates duplicate CONTACT statement definitions for individuals. GROUP statements can be coded anywhere in the INFORM policy member except between an INFORM statement and its corresponding CONTACT statements.

### GROUP



*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 can contain other group names only if those groups were previously defined. A group cannot refer to itself.

#### Example:

This example defines groups so that contact statements are not duplicated in multiple policies. PersonC and CompanyA are listed in multiple groups:

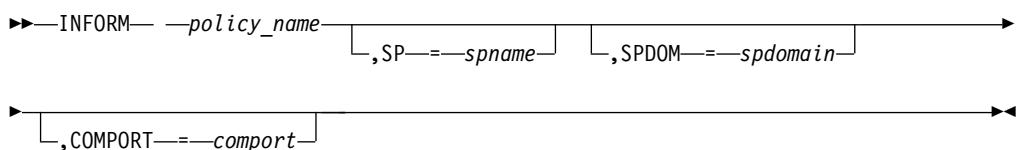
```
GROUP DAYOPS,LIST=PERSONA,PERSONB,COMPANYA;
GROUP NITEOPS,LIST=COMPANYA,PERSONC;
GROUP WEEKENDS,LIST=PERSONC,PERSOND;
```

---

## 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 are to be notified. An INFORM statement must be followed by one or more CONTACT statements.

### INFORM



*policy\_name*

The policy name used to identify the list of contacts that 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.

**COMPRT**

The communications serial port to be used on the service point unless overridden by the individual contact statements. COMPRT 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 NT01I002. The service point domain defaults to the current domain and the COMPRT parameter is specified on the required contact statements that follow (not shown here).

INFORM COMPANYA, SP=NT01I001;

---

## 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.

---

## Chapter 6. Event/Automation Service Definition Statements

The Event/Automation Service serves as a gateway for event data between the NetView for z/OS management environment, managers and agents that handle Event Integration Facility (EIF) events, 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 NetView alerts and messages into EIF events before forwarding the event data to a designated event server. The Event/Automation Service also converts NetView 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 NetView alert receiver PPI mailbox.

If the Event/Automation Service is started from the IHSAEVNT job, the Event/Automation Service definition statements are read from the following files (by default) :

- IHSAINIT (global initialization file)
- IHSAACFG (alert adapter service configuration file)
- IHSABCFG (confirmed alert adapter service configuration file)
- IHSAMCFG (message adapter service configuration file)
- IHSANCFG (confirmed message adapter service configuration file)
- IHSAECFG (event receiver service configuration file)
- IHSATCFG (trap-to-alert service configuration file)
- IHSAATCF (alert-to-trap service configuration file)

If the Event/Automation Service is started from a UNIX System Services shell (command line), the Event/Automation Service definition statements are read from the following files (by default):

- /etc/netview/global\_init.conf (global initialization file)
- /etc/netview/alert\_adpt.conf (alert adapter service configuration file)
- /etc/netview/confirm\_alert\_adpt.conf (confirmed alert adapter service configuration file)
- /etc/netview/message\_adpt.conf (message adapter service configuration file)
- /etc/netview/confirm\_message\_adpt.conf (confirmed 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

### Purpose

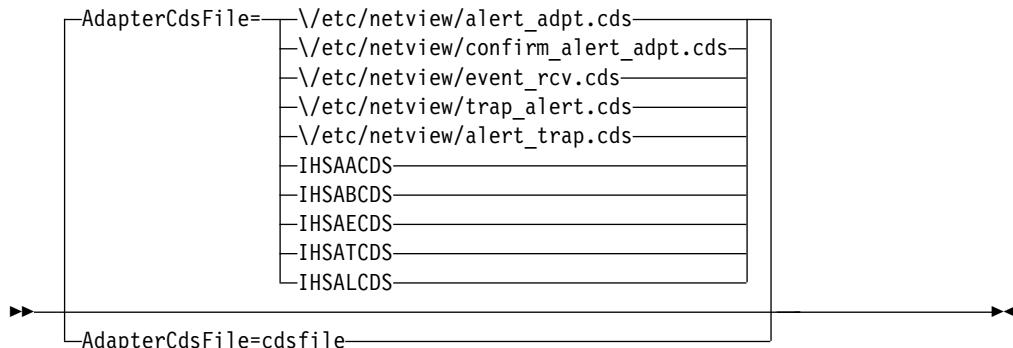
The AdapterCdsFile definition statement specifies the name of the Class Definition Statement (CDS) file. The AdapterCDSFile statement is used by the following services:

- Alert adapter service and the confirmed alert adapter service for converting alerts into Event Integration Facility (EIF) events
- Event receiver service for converting EIF events into alerts
- Trap-to-alert service for converting SNMP traps into alerts
- Alert-to-trap service for converting alerts into SNMP traps

### Syntax

The AdapterCdsFile statement has the following syntax:

#### AdapterCdsFile



where:

The *cdsfile* variable specifies the alert adapter, confirmed 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 it is 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, confirmed 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 not specify AdapterCdsFile for the alert adapter service, a default file name 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 confirmed alert adapter, a default file name of \\etc\\netview\\confirm\_alert\_adpt.cds is used if the

Event/Automation Service is started from a UNIX system services command line, or a default of IHSABCD 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 IHSACD 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 stops.

## Related statements

AdapterFmtFile

---

## AdapterFmtFile

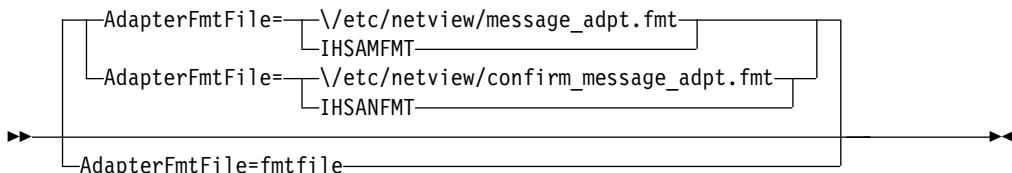
### Purpose

The AdapterFmtFile statement specifies the name of the format (FMT) file. The AdapterFmtFile statement is used by the message adapter service and the confirmed message adapter service for converting messages into Event Integration Facility (EIF) events.

### Syntax

The AdapterFmtFile statement has the following syntax:

#### AdapterFmtFile



*where:*

The *fmtfile* variable specifies the message adapter or confirmed message adapter FMT file. If preceded with the backslash (\) character, the AdapterFmtFilefilename is a complete MVS data set or an HFS file name. If it is 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 or confirmed 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 for the message adapter, a default file name 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 file name of IHSAMFMT is used if the Event/Automation Service is started from the IHSAEVNT job.
- If you do not specify the AdapterFmtFile statement for the confirmed message adapter, a default file name of \etc\netview\confirm\_message\_adpt.conf is used if the Event/Automation Service is started from a UNIX system services command line, or a default file name of IHSANFMT 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 stops.

## Related statements

AdapterCdsFile

---

# ALRTCFG

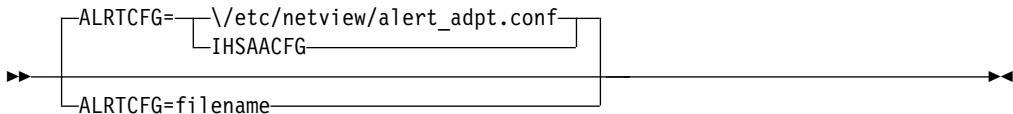
## Purpose

The ALRTCFG statement specifies the name of the alert adapter service configuration file.

## Syntax

The ALRTCFG statement has the following syntax:

### 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 it is 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 file name 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 file name of IHSAACFG is used if the Event/Automation Services 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 stops.
- This statement can be overridden with the ALRTCFG startup parameter. Refer to the *IBM 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

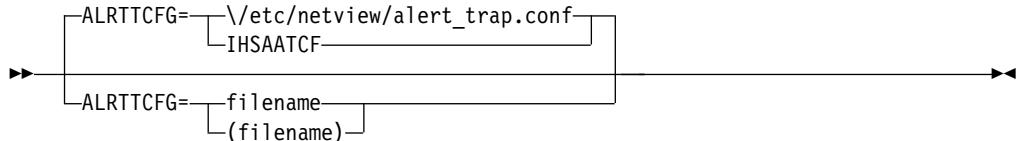
## Purpose

The ALRTTCFG statement specifies the name of the Alert to Trap service configuration file.

## Syntax

The ALRTTCFG statement has the following syntax:

### ALRTTCFG



*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 it is 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 file name of `\etc\netview\alert_trap.conf` is used if the Event/Automation Service is started UNIX system services command line, or a default file name 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 stops.

- This statement can be overridden with the ALRTTCFG startup parameter. Refer to the *IBM 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

### Purpose

The BufEvtMaxSize statement specifies the maximum size, in KB, of the message adapter service, confirmed message adapter service, alert adapter service, or confirmed alert adapter service event buffer file.

### Syntax

The BufEvtMaxSize statement has the following syntax:

#### **BufEvtMaxSize**



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, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

## Related statements

BufEvtPath, BufEvtRdBlklen, BufEvtShrinkSize, BufferEvents, BufferFlushRate, BufferEventsLimit, FilterCache.

---

## BufEvtNegRespLimit

### Purpose

The BufEvtNegRespLimit definition statement specifies the maximum number of negative responses that can be received for a particular Event Integration Facility (EIF) event by the confirmed alert or message adapter service. The negative response could be sent by either the primary or secondary server. A value of 0 indicates that no limit exists.

## Syntax

The BufEvtNegRespLimit statement has the following syntax:

### **BufEvtNegRespLimit**



*where:*

The *limit* variable specifies the maximum number of negative responses (a value from 0 - 128).

### Usage notes

- This statement is found in the confirmed message adapter service configuration file and the confirmed alert adapter service configuration file.
- A timeout caused by the inability to connect to an event server or because the event server does not respond, does not count towards the *limit*. These timeouts do not reset the negative response count that is maintained while attempting to send the event.
- When the limit is exceeded, the confirmed alert adapter (ALERTC) or confirmed message adapter (MESSAGEC) sends the IHS0025I message, which indicates that this condition occurred and logs the event for which it occurred.
- If the event is cached and the ALERTC task, the MESSAGEC task, or Event/Automation Service is restarted before the limit is reached, the negative response count is restarted at 0.
- If you do not specify the BufEvtNegRespLimit 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 stops.
- Consider the following items when you configure the ALERTC or MESSAGEC task:
  - Do not specify an event server in the list of servers specified on the ServerLocation statement. An event server will not send the confirmation that this adapter expects for each EIF event sent.
  - An EIF event is discarded when the negative response limit is exceeded. This is true even if the adapter has not made a complete pass through the ServerLocation list.
  - If BufferEvents=no is specified, an EIF event is discarded after one pass through the ServerLocation list (if the event is not successfully sent to and confirmed by any servers in the list).

---

## BufEvtPath

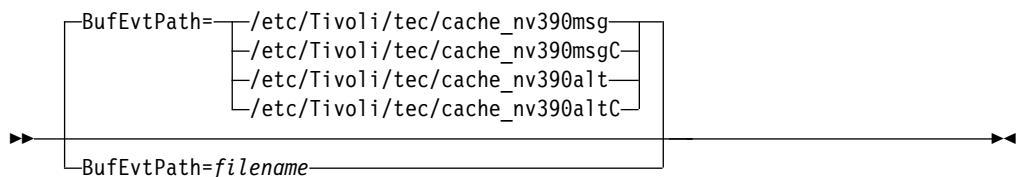
### Purpose

The BufEvtPath statement specifies the full path name of the message adapter service, confirmed message adapter service, alert adapter service, or confirmed alert adapter service event buffer file.

## Syntax

The BufEvtPath statement has the following syntax:

### **BufEvtPath**



*where:*

The *filename* variable specifies the full path name of the buffer event file.

### **Usage notes**

- This statement is found in the following configuration files:
  - message adapter service
    - path name: /etc/Tivoli/tec/cache\_nv390msg
  - confirmed message adapter service
    - path name: /etc/Tivoli/tec/cache\_nv390msgC
  - alert adapter service
    - path name: /etc/Tivoli/tec/cache\_nv390alt
  - confirmed alert adapter service
    - path name: /etc/Tivoli/tec/cache\_nv390altC
- If you do not specify the BufEvtPath statement, a default file name 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 stops.

### **Related statements**

BufEvtMaxSize, BufEvtRdBlklen, BufEvtShrinkSize, BufferEvents, BufferFlushRate, BufferEventsLimit, FilterCache.

---

## **BufEvtRdBlklen**

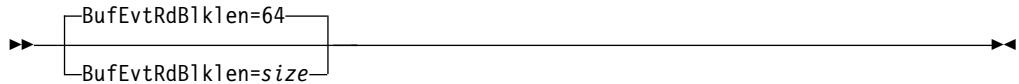
### **Purpose**

The BufEvtRdBlklen definition statement specifies the size, in KB, of the buffer used by the message adapter, confirmed message adapter, alert adapter, and confirmed alert adapter services to read data from the event buffer. Data is read from the event buffer in data blocks of the specified size.

## Syntax

The BufEvtRdBlklen statement has the following syntax:

## **BufEvtRdBklen**



*where:*

The *size* variable specifies the size in KB.

### **Usage notes**

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- If you do not specify the BufEvtRdBklen definition statement, a default of 64 is used.
- If any single event in the event buffer is larger than the BufEvtRdBklen, 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**

### **Purpose**

The BufEvtShrinkSize definition statement specifies the amount, in KB, to shrink the message adapter, confirmed message adapter, alert adapter, or the confirmed alert adapter service event buffer file when BufEvtMaxSize is exceeded.

### **Syntax**

The BufEvtShrinkSize statement has the following syntax:

## **BufEvtShrinkSize**



*where:*

The *amount* variable specifies the amount.

### **Usage notes**

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- If you do not specify the BufEvtShrinkSize statement, a default of 8 is used.

- If the event buffer is to 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 stops.
- 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

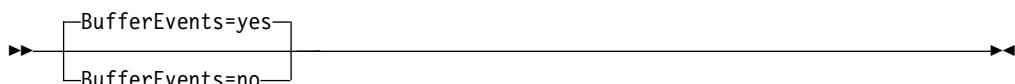
## Purpose

The `BufferEvents` definition statement specifies whether event buffering is enabled for the message adapter, confirmed message adapter, alert adapter, or confirmed alert adapter service.

## Syntax

The `BufferEvents` statement has the following syntax:

### **BufferEvents**



*where:*

**yes** | YES specifies that event buffering is enabled.

**no** | NO specifies that event buffering is disabled.

## Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- If you do not specify the `BufferEvents` statement, a default of YES is used.
- If you specify an incorrect value, event buffering is 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

### Purpose

The BufferEventsLimit definition statement specifies the maximum number of events that can be buffered during the current connection outage for the message adapter, confirmed message adapter, alert adapter, or confirmed alert adapter service. A value of 0 indicates that no limit exists.

### Syntax

The BufferEventsLimit statement has the following syntax:

#### BufferEventsLimit



where:

The *limit* variable specifies the maximum number of events.

### Usage notes

- Whenever an event is successfully sent to a designated event 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, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

### Related statements

BufEvtMaxSize, BufEvtPath, BufEvtRdBlklen, BufEvtShrinkSize, BufferEvents, BufferFlushRate, FilterCache.

---

## BufferFlushRate

### Purpose

The BufferFlushRate definition statement specifies the number of buffered events sent per minute by the message adapter, confirmed message adapter, alert adapter, or confirmed alert adapter service when a lost connection has been recovered.

### Syntax

The BufferFlushRate statement has the following syntax:

## **BufferFlushRate**



**where:**

The *rate* variable specifies the number of events per minute.

### **Usage notes**

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

### **Related statements**

BufEvtMaxSize, BufEvtPath, BufEvtRdBlklen, BufEvtShrinkSize, BufferEvents, BufferEventsLimit, FilterCache.

---

## **CALRTCFG**

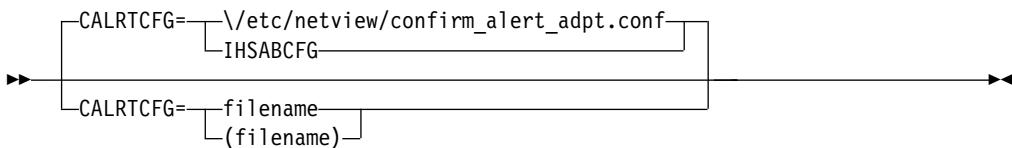
### **Purpose**

The CALRTCFG statement specifies the name of the confirmed alert adapter service configuration file.

### **Syntax**

The CALRTCFG statement has the following syntax:

## **CALRTCFG**



**where:**

The *filename* variable specifies the confirmed alert adapter service configuration file. If preceded with the backslash (\), the file name is a complete MVS data set or an HFS file name.

### **Usage notes**

- This statement is found in the global initialization file.
- If you specify a complete file name, the backslash must immediately precede the complete file name; there can be no intervening spaces.

- If you do not specify the CALRTCFG statement, a default file name of `\etc\netview\confirm_alert_adpt.conf` is used if the Event/Automation Service is started from a UNIX System Services command line, or a default file name of IHSABCFG is used if the Event/Automation Services 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 stops.
- This statement can be overridden with the CALRTCFG startup parameter. Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

## Related statements

ALRTTCFG, CMSGCFG, ERCVCFG, MSGCFG, TALRTCFG

## CMSGCFG

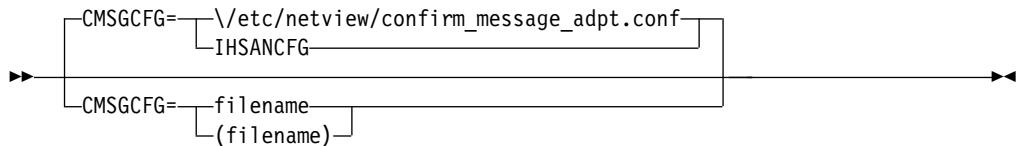
### Purpose

The CMSGCFG statement specifies the name of the confirmed message adapter service configuration file.

### Syntax

The CMSGCFG statement has the following syntax:

#### CMSGCFG



*where:*

The *filename* variable specifies the confirmed message adapter service configuration file. If *filename* is preceded with the backslash (\), the file name is a complete MVS data set or an HFS file name. If *filename* is not preceded with the backslash, 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 must immediately precede the complete file name; there can be no intervening spaces.
- If you do not code the CMSGCFG statement, a default file name of `\etc\netview\confirm_message_adpt.conf` is used if the Event/Automation Service is started from a UNIX System Services command line, or a default file name of IHSANCFG is used if the Event/Automation Services 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 stops.

- This statement can be overridden with the CMSGCFG startup parameter. Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

## Related statements

ALRTCFG, ALRTTCFG, CALRTCFG, ERCVCFG, MSGCFG, TALRTCFG

---

## Community

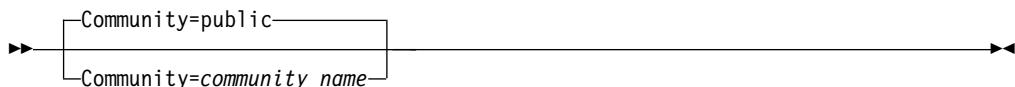
### Purpose

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.

### Syntax

The Community statement has the following syntax:

#### Community



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 stops.

---

## ConnectionMode

### Purpose

For the message adapter service or the alert adapter service, the ConnectionMode statement specifies how to connect to the designated event server. For the confirmed message adapter service or the confirmed alert adapter service, this statement specifies how to connect to an event server.

## Syntax

### ConnectionMode



The ConnectionMode statement has the following syntax:

*where:*

**connection\_less** specifies that a new IP socket connection is to be established for each Event Integration Facility (EIF) event that is to be sent to the server.

**connection\_oriented** or **CO** specifies that the IP socket connection is retained between sending EIF events to the server.

### Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

### Related statements

RetryInterval.

---

## Enterpriseoid

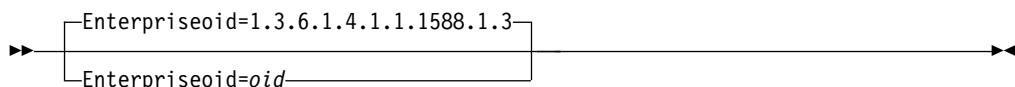
### Purpose

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.

## Syntax

The Enterpriseoid statement has the following syntax:

### Enterpriseoid



*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 object ID, 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 stops.

---

## ERCVCFG

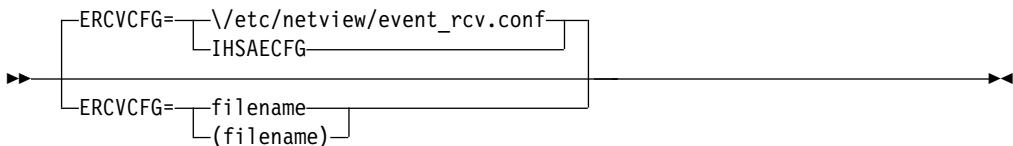
### Purpose

The ERCVCFG statement specifies the name of the Event Receiver service configuration file.

### Syntax

The ERCVCFG statement has the following syntax:

#### ERCVCFG



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 it is 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 file name 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 file name of IHSAECFG is used if the Event/Automation Services 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 stops.
- This statement can be overridden with the ERCVCFG startup parameter. Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

## Related statements

ALRTCFG, MSGCFG, TALRTCFG, ALRTTCFG

---

## EventMaxSize

### Purpose

The EventMaxSize statement specifies the maximum length of the Event Integration Facility (EIF) events that is generated by the message adapter service, confirmed message adapter service, alert adapter service, or confirmed alert adapter service.

### Syntax

The EventMaxSize statement has the following syntax:

#### EventMaxSize



where:

The *size* variable specifies the maximum size of an event in bytes.

### Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

---

## FallbackValue

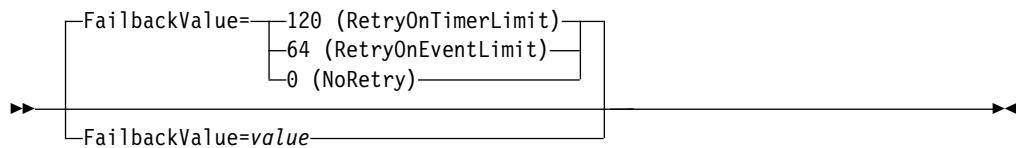
### Purpose

The FallbackValue statement qualifies the retry condition for the ServerFallback statement. If RetryOnTimerLimit is specified on the ServerFallback statement, this statement specifies the amount of time in seconds before fail back. If RetryOnEventLimit is specified on the ServerFallback statement, this statement specifies the number of events before failback. If NoRetry is specified on the ServerFallback statement, this statement has no effect.

### Syntax

The FallbackValue statement has the following syntax:

#### FallbackValue



*where:*

The *value* specifies either the number of events (RetryOnEventLimit) or the number of seconds (RetryOnTimerLimit) to wait before failback to the first server in the ServerLocation list.

### Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- If you do not specify the FailbackValue definition statement, a default value of 0 is used for a ServerFallback value of NoRetry. A default value of 64 events is used for a ServerFallback value of RetryOnEventLimit. A default value of 120 seconds is used for a ServerFallback value of RetryOnTimerLimit.
- If you specify a value that is not valid, an error message is issued and the specific service stops.

### Related statements

ServerFallback, ServerLocation, RetryInterval

---

## Filter

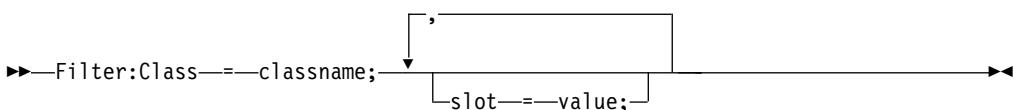
### Purpose

The filter statement specifies the criteria used to filter events for the message adapter service, confirmed message adapter service, alert adapter service, or confirmed alert adapter service. The filter can be used either to pass events to the designated event server, or to prevent events from being sent to the designated event server, depending on the FilterMode statement.

### Syntax

The filter statement has the following syntax:

#### Filter



*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, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.

- 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. When a match is made, the FileMode 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 stops.

## Related statements

FilterMode, FilterCache

## FilterCache

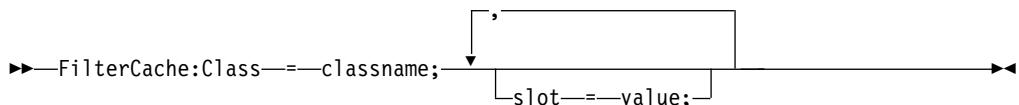
## Purpose

The FilterCache statement specifies the criteria used to filter events for event buffering for the message adapter service, confirmed message adapter service, alert adapter service, or confirmed 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 FileMode statement.

## Syntax

The FilterCache statement has the following syntax:

## FilterCache



*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, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
  - Event Integration Facility (EIF) events that are discarded because of 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 is not buffered.

- Multiple FilterCache statements can be specified in the configuration file. An attempt to match on each statement is made until a match is made or all FilterCache statements are exhausted.
- If you specify a value that is not valid, an error message is issued and the specific service stops.

## Related statements

Filter, BufferEvents

---

## FilterMode

### Purpose

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, confirmed message adapter service, alert adapter service, or confirmed alert adapter service.

### Syntax

The FilterMode statement has the following syntax:

#### FilterMode



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, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

## Related statements

Filter, FilterCache, BufferEvents

---

## Hostname

### Purpose

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.

### Syntax

The Hostname statement has the following syntax:

#### Hostname



*where:*

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 *loopback* is used.
- If you specify a value that is not valid, an error message is issued and the alert-to-trap service stops.

---

## MSGCFG

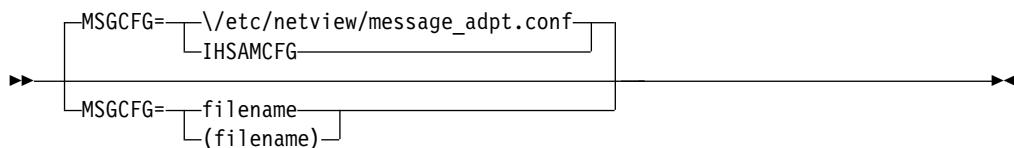
### Purpose

The MSGCFG statement specifies the name of the message adapter service configuration file.

### Syntax

The MSGCFG statement has the following syntax:

#### 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 it is 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 file name 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 file name of IHSAMCFG is used if the Event/Automation Services 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 stops.
- This statement can be overridden with the MSGCFG startup parameter. Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

### Related statements

ALRTCFG, CALRTCFG, CMSGCFG, ERCVCFG, TALRTCFG, ALRTTCFG

---

## NetViewAlertReceiver

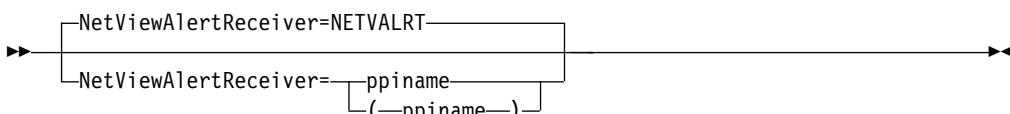
### Purpose

The NetViewAlertReceiver statement specifies the PPI mailbox name that identifies the NetView alert receiver program.

### Syntax

The NetViewAlertReceiver statement has the following syntax:

#### NetViewAlertReceiver



*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 stops.

---

## NOSTART

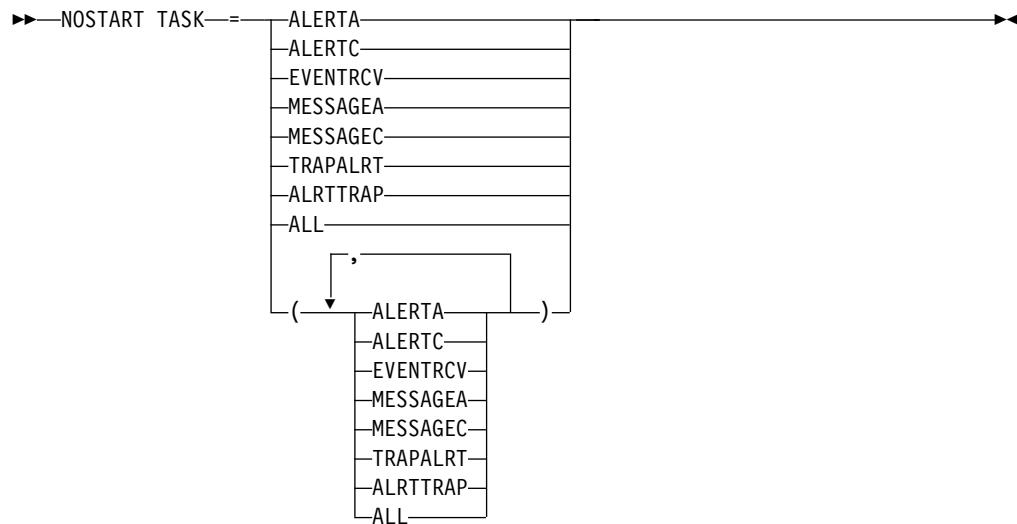
### Purpose

The NOSTART statement specifies which service tasks, if any, are *not* to be started when the Event/Automation Service is initialized.

### Syntax

The NOSTART statement has the following syntax:

#### **NOSTART**



*where:*

#### **ALERTA**

specifies that the alert adapter service is not to be started

#### **ALERTC**

specifies that the confirmed alert adapter service is not to be started

#### **EVENTRCV**

specifies that the event receiver service is not to be started

#### **MESSAGEA**

specifies that the message adapter service is not to be started

#### **MESSAGEC**

specifies that the confirmed message adapter service is not to be started

#### **TRAPALRT**

specifies that the trap-to-alert service is not to be started

#### **ALRTTRAP**

specifies that the alert-to-trap service is not to be started

#### **ALL**

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 stops.

## OUTPUT

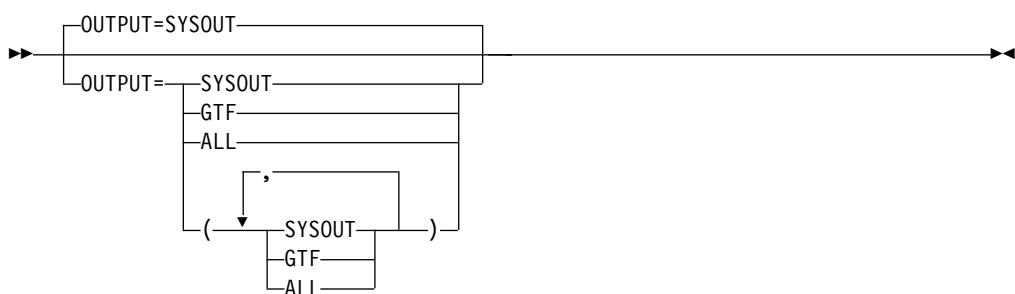
### Purpose

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.

### Syntax

The OUTPUT statement has the following syntax:

#### OUTPUT



*where:*

#### SYSOUT

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 the Event/Automation Service attempts to send trace/error data to the GTF, a single warning message is issued. When the GTF is activated, the Event/Automation Service begins sending trace or error data to it. Any data sent to GTF while it is inactive is lost, unless the data 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 stops.

- This statement can be overridden with the OUTPUT modification command. Refer to the *IBM Tivoli NetView for z/OS Command Reference Volume 2 (O-Z)* or the NetView online help for more information about Event/Automation Service commands.

---

## PortNumber

### Purpose

The PortNumber statement specifies the port number that the event receiver and trap-to-alert services are to use.

### Syntax

The PortNumber statement has the following syntax:

#### PortNumber



where:

The *port* variable specifies a valid IP port number. If zero (0), this indicates that the service uses 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 stops.
- If you specify a value that is not valid, an error message is issued and the event receiver service stops.

### Related statements

UsePortmapper.

---

## PPI

### Purpose

The PPI statement specifies the PPI mailbox name that identifies this Event/Automation Service

### Syntax

#### PPI



*where:*

The *ppiname* variable specifies a 1 - 8 character PPI mailbox name for the Event/Automation Service.

### 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 stops.
- This statement can be overridden with the PPI startup parameter. Refer to the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

## RetryInterval

### Purpose

For the message adapter or the alert adapter service, the RetryInterval statement specifies how many seconds the service waits before attempting to reestablish a TCP/IP session with an event server. This statement only applies if the ConnectionMode statement is specified as connection\_oriented.

For the confirmed message adapter or the confirmed alert adapter service, the RetryInterval statement specifies how many seconds the service waits before attempting to reestablish a TCP/IP session with an event server. This statement only applies if the ConnectionMode statement is specified as connection\_oriented. For the confirmed adapters, the RetryInterval statement also specifies the amount of time in seconds that a confirmed adapter waits for a reply after an Event Integration Facility (EIF) event is successfully sent. In this case, the RetryInterval statement applies if the ConnectionMode statement is specified as connection\_less or connection\_oriented.

### Syntax

The RetryInterval statement has the following syntax:

#### RetryInterval



*where:*

The *interval* variable specifies the number of seconds to wait.

## Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

## Related statements

ConnectionMode

---

## ServerFallback

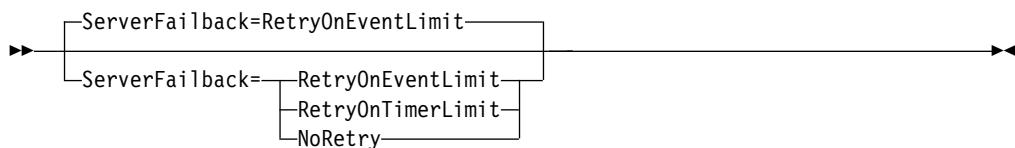
### Purpose

The ServerFallback statement controls how to retry servers after a server fail over occurs. A server failover occurs whenever a connection cannot be made to the first (also called primary) server in the ServerLocation list.

### Syntax

The ServerFallback statement has the following syntax:

#### ServerFallback



*where:*

#### RetryOnEventLimit

specifies that servers in the ServerLocation list are retried after a number of events specified on the FallbackValue statement. The retry only occurs if the last connection was to a non-primary server

#### RetryOnTimerLimit

specifies that servers in the ServerLocation list are retried after a number of seconds specified on the FallbackValue statement have elapsed. The retry only occurs if the last connection was to a non-primary server

#### NoRetry

specifies that the connection stays with the non-primary server until a manual reset is issued with the RESETSRV command

## Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- If you do not specify the ServerFallback statement, a default of RetryOnEventLimit is used.

- If you specify a value that is not valid, an error message is issued and the specific service stops.
- If you are using a ConnectionMode of connection\_oriented, when a connection is lost, servers in the ServerLocation list are retried from the beginning of the list until a connection is made or until the list is exhausted. A lost connection is a reset condition in connection\_oriented mode. A ConnectionMode of connection\_less does not interpret a lost connection as a reset condition, because the normal state of a connection in connection\_less mode is disconnected.

## Related statements

[FallbackValue](#), [ServerLocation](#), [RetryInterval](#)

# ServerLocation

## Purpose

For the message adapter and alert adapter services, the ServerLocation statement specifies the name of the host on which the designated event server is installed. This is used to forward converted events.

For the confirmed message adapter and confirmed alert adapter services, the first value on the ServerLocation statement specifies the primary event server, and the other values are secondary event servers that are used when a connection cannot be established or a response is not received from a primary event server.

## Syntax

The ServerLocation statement has the following syntax:

### ServerLocation



```
>> ServerLocation = [serverlist]
 ↓
 servername
```

where:

For the message adapter and alert adapter services, the *servername* variable specifies the location of the designated event server, and must use one of the following formats:

- A TCP/IP host name or TCP/IP address. This form is used if the adapters are using a nonsecure connection to the event server
- An HFS fully qualified file name if the TestMode statement indicates that Event Integration Facility (EIF) events are to be sent to a file for debug purposes.

For the confirmed message adapter and alert adapter services, the *servername* variable specifies the location of the event server, and must use one of the following formats:

- A TCP/IP host name or TCP/IP address. This form is used if the confirmed adapters are using a nonsecure connection to the event server
- An HFS fully qualified file name if the TestMode statement indicates that EIF events are to be sent to a file for debug purposes.

Note that the confirmed adapters can connect to servers in an IPv6 network.

### Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- The ServerLocation statement must be coded.
- The ServerLocation statement can contain up to 16 comma-separated values. For the message adapter and alert adapter services, the first location is the primary event server, the other locations are secondary event servers that are to be used only when a connection cannot be established to the primary event server. For the confirmed message adapter and confirmed alert adapter services, the first value is considered a primary event server, and the other values are considered secondary event servers.
- If you specify a value that is not valid, an error message is issued and the specific service stops.

### Related statements

ServerPort

---

## ServerPort

### Purpose

The ServerPort statement specifies the port numbers on which the hosts specified in the ServerLocation listen. This is used by the message adapter, confirmed message adapter, alert adapter, and confirmed alert adapter services for forwarding converted Event Integration Facility (EIF) events.

### Syntax

The ServerPort statement has the following syntax:

#### ServerPort



#### Notes:

- 1 alert adapter service and message adapter service
- 2 confirmed alert adapter service and confirmed message adapter service

#### 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.
- This statement is found in the confirmed alert adapter service configuration file and the confirmed message adapter service configuration file.
- For the alert adapter service and the message adapter service, the default value is zero (0). A value of 0 indicates that PortMapper is used to retrieve the port numbers of the corresponding event server.
- For the confirmed alert adapter service and the confirmed message adapter service, the default value is 5539. Do not code a value of 0 because the target event server will not register its port.
- 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 stops.

## Related statements

ServerLocation

---

## TALRTCFG

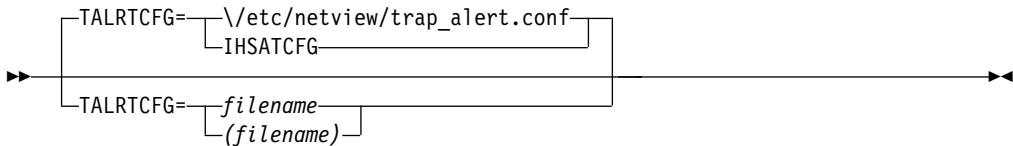
### Purpose

The TALRTCFG statement specifies the name of the Trap to Alert service configuration file.

### Syntax

The TALRTCFG statement has the following syntax:

#### TALRTCFG



*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 file name of `\etc\netview\trap_alert.conf` is used if the Event/Automation Service is started

from the UNIX system services command line, or a default file name 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 stops.
- This statement can be overridden with the TALRTCFG startup parameter. See the *IBM Tivoli NetView for z/OS Installation: Configuring Additional Components* for more information on Event/Automation Service startup parameters.

## Related statements

ALRTCFG, MSGCFG, ERCVCFG, ALRTTCFG

---

## TestMode

### Purpose

The TestMode statement specifies whether Event Integration Facility (EIF) events are sent to an event server or are sent to a debug file for the message adapter, confirmed message adapter, alert adapter, or confirmed alert adapter service. The debug file is specified in the ServerLocation statement if the test mode is enabled.

### Syntax

The TestMode statement has the following syntax:

#### TestMode



where:

**yes | YES**

specifies that test mode is enabled. All events are routed to the debug file

**no | NO**

specifies that test mode is disabled

### Usage notes

- This statement is found in the message adapter service, confirmed message adapter service, alert adapter service, and confirmed alert adapter service configuration files.
- 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 stops.

---

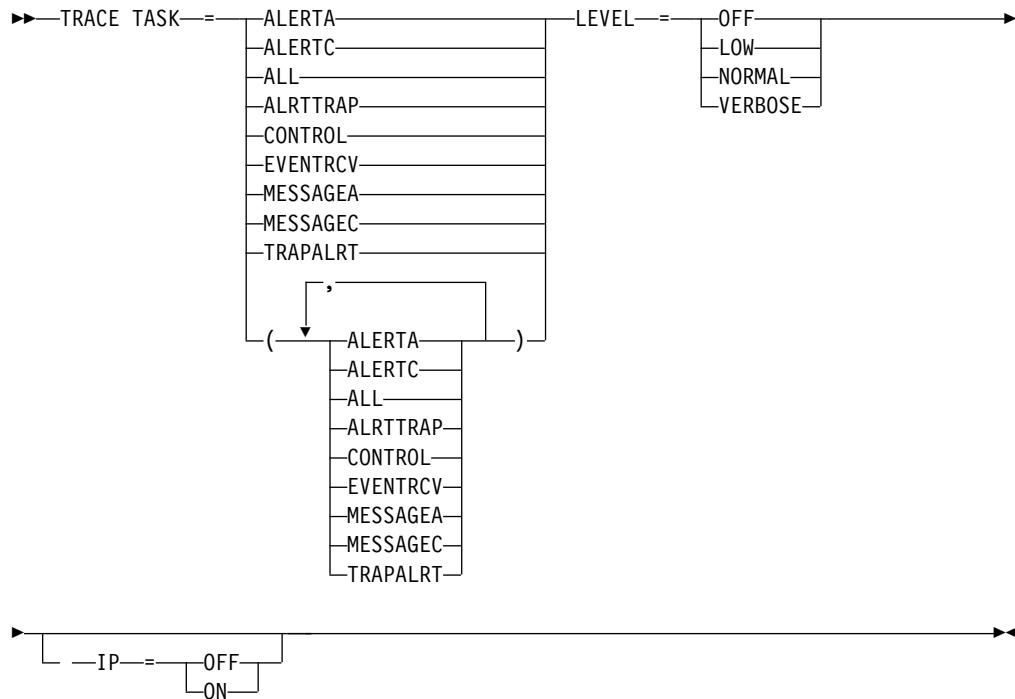
## TRACE

### Purpose

The TRACE statement specifies what level of tracing to enable for one or more service tasks.

## Syntax

The TRACE statement has the following syntax:



*where:*

## **ALERTA**

specifies that the alert adapter task is traced

ALERTC

specifies that the confirmed alert adapter task is traced

ALL

specifies that all tasks are traced

ALRTTRAP

specifies that the alert-to-trap service is traced

## **CONTROL**

specifies that the control task is traced

EVENTRCV

specifies that the event receiver task is traced.

## MESSAGEA

specifies that the message adapter task is traced

## MESSAGEC

specifies that the confirmed message adapter task is traced

TRAPALRT

specifies that the trap-to-alert service is traced

OFF

specifies that tracing for the specified tasks is disabled

**LOW**

specifies that the lowest level of tracing is enabled. Typically, this traces functions entry and exit

**NORMAL**

specifies the typical level of tracing is enabled. This traces the LOW level, plus any function flow information that aids problem determination

**VERBOSE**

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, GTF, or both.

**IP** specifies that the tracing of IP connection data is 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 stops.
- 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 *IBM Tivoli NetView for z/OS Command Reference Volume 1 (A-N)* for more information on Event/Automation Service commands.

---

## TruncateSV31s

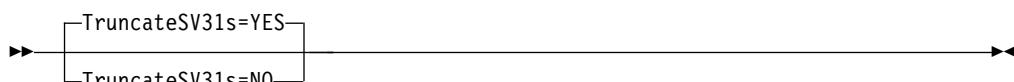
### Purpose

The TruncateSV31s statement controls the number of Self-defining Text Message subvectors (SV31s) that are created for large Event Integration Facility (EIF) event slot/value pairs or SNMP trap variable bindings.

### Syntax

The TruncateSV31s statement has the following syntax:

#### TruncateSV31s



*where:*

**YES** or **yes** specifies that incoming event data is limited to a single SV31. The incoming data is truncated if it is too large to fit into a single SV31.

**NO** or **no** specifies that incoming event data is placed into as many SV31s as are required to contain the data.

**Usage notes**

- This statement is found in the event receiver service and trap-to-alert services configuration files.
- If you do not specify the TruncateSV31s statement, a default of YES is used.

- If you specify a value that is not valid, an error message is issued and the service ends.

---

## UsePortMapper

### Purpose

The UsePortMapper statement specifies whether the event receiver service is to register itself with the PortMapper program. If the port number supplied on the PortNumber statement is zero (0), PortMapper is also used to assign a port to the event receiver service.

### Syntax

The UsePortMapper statement has the following syntax:

#### UsePortMapper



*where:*

yes | YES specifies the PortMapper is to be used.

no or NO 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 stops.

### Related statements

PortNumber

---

## Chapter 7. Correlation Service Definition Statements

The event correlation service runs under UNIX Systems Services and outside the NetView address space to process events. The connection between the correlation service and the NetView program is made through TCP/IP sockets. For more information, refer to the *IBM Tivoli NetView for z/OS Automation Guide*.

The event correlation service definition statements are located in the following file:  
`/etc/netview/v6r2m1/properties/correlator.properties`

These statements are system controlling constants that are read when the event correlation service is initialized.

---

### CLIENTPORT

#### Purpose

The CLIENTPORT statement specifies the port number that the correlation service uses to connect to the Common Event Infrastructure client.

#### Syntax

##### CLIENTPORT



The syntax for the CLIENTPORT statement follows:

*where:*

*port*

Specifies a valid IP port number.

#### Usage notes

- If you do not code the CLIENTPORT statement, a default of 4054 is used.
- If you are using Common Event Infrastructure support, the port number must be the same as the port number specified on the LCLPORT statement in the client property file.
- If you are not using Common Event Infrastructure support, this statement is ignored.

---

### CLIENTLISTPORT

#### Purpose

The CLIENTLISTPORT statement specifies the port number that the correlation service uses for requests from the Common Event Infrastructure client.

## Syntax

The syntax for the CLIENTLISTPORT statement follows:

### CLIENTLISTPORT



*where:*

*port*

Specifies a valid IP port number.

### Usage notes

- If you do not code the CLIENTLISTPORT statement, a default of 4053 is used.
- If you are using Common Event Infrastructure support, the port number must be the same as the port number specified on the NVPORt statement in the client property file.
- If you are not using Common Event Infrastructure support, this statement is ignored.

---

## JLOG

### Purpose

The JLOG.CONFIGURATION statement specifies the location of the configuration files used for the correlation logging service.

### Syntax

The syntax for the JLOG statement follows:



*where:*

*file\_name*

Specifies the location of the logging service.

### Usage notes

- The default logging service is located in the following file:  
/etc/netview/v6r2m1/properties/jlog.properties
- If you specify a value that is not valid, an error message is issued and the correlation service ends.
- You can set logging levels by using the jlog.category.com.tivoli.nv390.correlator property.

---

## LCLPORT

### Purpose

The LCLPORT statement specifies the port number that the correlation service uses for requests from the NetView program.

### Syntax

The syntax for the LCLPORT statement follows:

#### LCLPORT



*where:*

*port*

Specifies a valid IP port number.

### Usage notes

- If you do not code the LCLPORT statement, a default of 4051 is used.
- The port number must be the same as the port number that is specified on the CORRELATION (SERVERPORT) statement in the CNMSTYLE member. For more information about the CORRELATION statement, see “CORRELATION” on page 56.

---

## NVHOST

### Purpose

The NVHOST statement specifies the TCP name for the NetView host to which error messages from the Common Event Infrastructure client are forwarded. In most cases, this is the local NetView host.

### Syntax

The syntax for the NVHOST statement follows:

#### NVHOST



*where:*

*hostname*

Specifies a valid IP address or server name.

### Usage notes

- If you do not code the NVHOST statement, a default of localhost is used.

- If you are using Common Event Infrastructure support, the *hostname* must match the NVHOST property in the client property file.
- If you are not using Common Event Infrastructure support, this statement is ignored.

## NVLOCATE

### Purpose

The NVLOCATE statement defines which language support is used with the NetView host.

### Syntax

#### NVLOCATE



The syntax for the transTbl statement is:

*where:*

#### DEFAULT

Selects the same language as the correlator service host. This is the default.

#### ENGLISH

Selects the English language EBCDIC NetView host.

#### KANJI

Selects the Japanese language EBCDIC NetView host.

## NVPORT

### Purpose

The NVPORT statement specifies the port number that the NetView program uses for responses from the correlation service.

### Syntax

The syntax for the NVPORT statement follows:

#### NVPORT



*where:*

*port*

Specifies a valid IP port number.

## Usage notes

- If you do not code the NVPORT statement, a default of 4050 is used.
- The port number must be the same as the port number that is specified on the CORRELATION (LOCALPORT) statement in the CNMSTYLE member. For more information about the CORRELATION statement, see "CORRELATION" on page 56.

---

## PURGETIMER

### Purpose

The PURGETIMER statement specifies the amount of time that entries are kept on the retry queues.

### Syntax

The syntax for the PURGETIMER statement is:

#### PURGETIMER



*where:*

*seconds*

Specifies the number of seconds.

### Usage notes

- If you do not code the PURGETIMER statement, a default of 1800 seconds is used.
- Because of the algorithm used, the actual time entries are kept on the retry queues and might be the sum of the times specified on the PURGETIMER and RETRYTIME statements.

---

## QUELIMIT

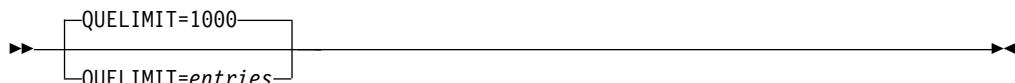
### Purpose

Use the QUELIMIT statement to define the number of entries to keep on the retry queue.

### Syntax

The syntax for the QUELIMIT statement is:

#### QUELIMIT



*where:*

*entries*

Specifies the number of entries to keep on the retry queue.

**Usage notes**

- If you do not code the QUELIMIT statement, a default of 1000 entries is used.

---

## RETRYTIME

### Purpose

The RETRYTIME statement specifies the interval of time (in seconds) between retries to process an entry on the queue.

### Syntax

The syntax for the RETRYTIME statement is:

#### RETRYTIME



*where:*

*seconds*

Specifies the number of seconds to wait.

**Usage notes**

- If you do not code the RETRYTIME statement, a default of 5 seconds is used.

---

## RULEFILE

### Purpose

The RULEFILE statement specifies the location of the default rules base used by the correlation service.

### Syntax

The syntax for the RULEFILE statement follows:



*where:*

*file\_name*

Specifies the location of the correlation service rules base.

**Usage notes**

- The default rules base is located in the following file:  
/var/netview/v6r2m1/rulefiles/znrules.xml

---

## Chapter 8. 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

#### Purpose

Use the ASYNC\_TASKS statement 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.

#### Syntax

The ASYNC\_TASKS statement has the following syntax:

#### ASYNC\_TASKS



*where:*

5|nn

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

#### Purpose

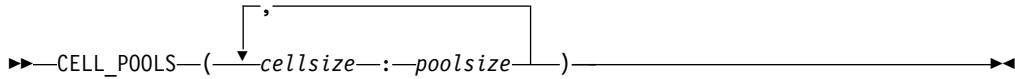
Use the CELL\_POOLS statement to define the number and size of cell pools. Storage is allocated from the operating system in large blocks (cell pools) and is subdivided by RODM. The cell pools are divided into multiple elements of the same size. You can specify CELL\_POOLS statements to identify cell size and pool size combinations for up to 200 cell sizes in addition to the default RODM cell sizes.

#### Syntax

The CELL\_POOLS statement has the following syntax:

## **CELL\_POOLS**

```
►►CELL_POOLS-(cellsize:poolsize)►►
```



*where:*

### *cellsize*

Specifies the cell size of each pool, in bytes. You can specify a value in the range of 8 bytes to 512 kilobytes (524288). 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 do not specify any CELL\_POOLS statements, or if you specify a CELL\_POOLS statement with no values, RODM uses only the default cell and cell pool values. The following values are the 56 default values for cell pools:

| Cell Size<br>(bytes) | Pool Size<br>(pages) | Cell Size<br>(bytes) | Pool Size<br>(pages) |
|----------------------|----------------------|----------------------|----------------------|
| 8                    | 1                    | 176                  | 4                    |
| 12                   | 1                    | 184                  | 4                    |
| 16                   | 1                    | 192                  | 4                    |
| 20                   | 1                    | 200                  | 4                    |
| 24                   | 1                    | 208                  | 4                    |
| 28                   | 1                    | 216                  | 4                    |
| 32                   | 1                    | 224                  | 4                    |
| 36                   | 2                    | 232                  | 4                    |
| 40                   | 2                    | 240                  | 4                    |
| 48                   | 2                    | 248                  | 4                    |
| 52                   | 2                    | 256                  | 4                    |
| 56                   | 2                    | 384                  | 6                    |
| 60                   | 2                    | 484                  | 9                    |
| 64                   | 2                    | 512                  | 8                    |
| 68                   | 3                    | 768                  | 9                    |
| 72                   | 3                    | 1024                 | 8                    |
| 80                   | 3                    | 1536                 | 12                   |
| 88                   | 3                    | 2048                 | 16                   |
| 100                  | 3                    | 3072                 | 24                   |
| 104                  | 3                    | 4096                 | 32                   |
| 112                  | 3                    | 6144                 | 48                   |
| 120                  | 3                    | 8192                 | 64                   |
| 128                  | 3                    | 12288                | 96                   |
| 136                  | 4                    | 16384                | 128                  |

| Cell Size<br>(bytes) | Pool Size<br>(pages) | Cell Size<br>(bytes) | Pool Size<br>(pages) |
|----------------------|----------------------|----------------------|----------------------|
| 144                  | 4                    | 24576                | 192                  |
| 152                  | 4                    | 32768                | 256                  |
| 160                  | 4                    | 262144               | 64                   |
| 168                  | 4                    | 8388608              | 2048                 |

- The default cell pool definitions work well in most environments using the code that is supplied with RODM. However, when many objects are defined (more than approximately 130,000), RODM storage can become fragmented. To avoid excessive fragmentation, add the following statement to your RODM customization member:  

```
CELL_POOLS (524288)
```
- 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.

## Examples

The following example shows 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, 384:6)
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)
CELL_POOLS (524288)
```

## CHARACTER\_VALIDATION

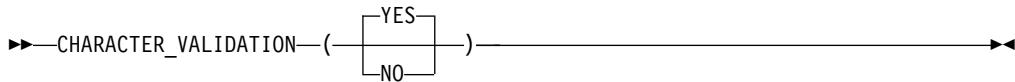
### Purpose

Use the CHARACTER\_VALIDATION statement to control the amount of validity checking RODM does for class, object, and field names.

### Syntax

The CHARACTER\_VALIDATION statement has the following syntax:

## **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, can 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: # @ . , : ; ? ( ) ' " - \_ & + % \* = < > /

For a complete description of valid class, object, and field names, refer to the *IBM Tivoli NetView for z/OS Resource Object Data Manager and GMFHS Programmer's Guide*.

**NO** 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 *IBM 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**

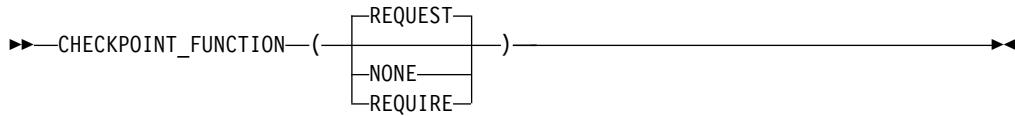
### **Purpose**

Use the CHECKPOINT\_FUNCTION statement to control the RODM checkpoint function.

### **Syntax**

The CHECKPOINT\_FUNCTION statement has the following syntax:

## **CHECKPOINT\_FUNCTION**



*where:*

### **REQUEST**

Specifies that the RODM checkpoint function is "requested to be enabled". With this setting, RODM continues to allocate storage for new objects and classes, even if that storage exceeds the capacity of the checkpoint data sets. If the storage that is allocated by RODM grows beyond the capacity of the checkpoint data sets, the checkpoint function is disabled and RODM continues. This is the default.

Additionally, this setting specifies that during a cold start initialization, RODM continues 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 (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 exceeds the capacity of the checkpoint data sets. This causes the capacity of RODM to be limited to the capacity of the checkpoint data sets.

Additionally, this option specifies that during a cold start initialization, RODM terminates if a checkpoint function setup failure occurs.

## **Usage notes**

- This is not a statement that can be reloaded. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

## **CONCURRENT\_USERS**

### **Purpose**

Use the CONCURRENT\_USERS statement 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.

### **Syntax**

The CONCURRENT\_USERS statement has the following syntax:

## **CONCURRENT\_USERS**



*where:*

**10|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

- Ensure that this number reflects 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 statement that can be reloaded . If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

---

## CONNECT\_VIOLATION\_MESSAGE

### Purpose

Use the CONNECT\_VIOLATION\_MESSAGE statement to specify whether RODM is to write message EKG2000I when it encounters a problem while processing a CONNECT request.

### Syntax

The CONNECT\_VIOLATION\_MESSAGE statement has the following syntax:

#### **CONNECT\_VIOLATION\_MESSAGE**

►►CONNECT\_VIOLATION\_MESSAGE—(NO|YES)—►►

*where:*

**NO|YES**

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

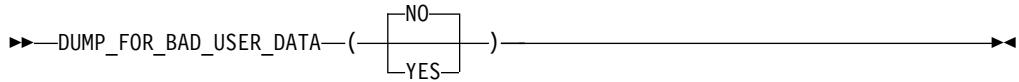
### Purpose

Use the DUMP\_FOR\_BAD\_USER\_DATA statement to specify whether RODM is dumped when it encounters a problem while copying user data.

### Syntax

The DUMP\_FOR\_BAD\_USER\_DATA statement has the following syntax:

## DUMP\_FOR\_BAD\_USER\_DATA



*where:*

**NO|YES**

Specifies whether RODM is to be dumped 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

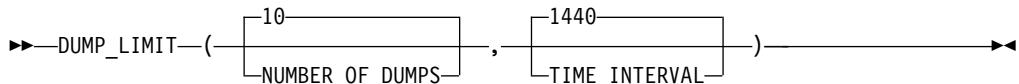
---

## DUMP\_LIMIT

Use the DUMP\_LIMIT statement 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 DUMP\_LIMIT statement has the following syntax:

### DUMP\_LIMIT



*where:*

**10|NUMBER\_OF\_DUMPS**

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.

**1440|TIME\_INTERVAL**

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 TIME\_INTERVAL is 60 minutes, another dump is possible if NUMBER\_OF\_DUMPS has not already been reached during the last hour.

If you do not want to limit the number of dumps, set NUMBER\_OF\_DUMPS to the maximum value (30), and TIME\_INTERVAL to the minimum value (1).

You can specify a value from 1–40320. The default is 1440 (one day).

### Usage Notes:

- When 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 DUMP\_LIMIT. Refer to the DMPRESET command for more information.
- This is a statement that can be reloaded. 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\_SCOPE

Use the DUMP\_SCOPE statement to specify which user address spaces are dumped when an ABEND occurs in RODM.

The DUMP\_SCOPE statement has the following syntax:

### DUMP\_SCOPE



*where:*

#### REQUESTOR

Specify REQUESTOR to dump the address space that is involved in the ABEND. This is the default.

#### CONNECTED

Specify CONNECTED to dump up to 10 address spaces that are connected to RODM, in addition to the requesting address space, and the data spaces.

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 are to be dumped.

#### 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.
- 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

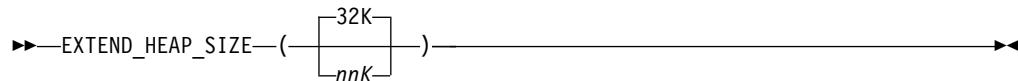
---

## **EXTEND\_HEAP\_SIZE**

Use the EXTEND\_HEAP\_SIZE statement to define the size, in bytes, of the extended PL/I heap storage.

The EXTEND\_HEAP\_SIZE statement has the following syntax:

### **EXTEND\_HEAP\_SIZE**



*where:*

**32K|nnK**

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 called 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 statement that can be reloaded. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

**Related Statements:** PLI\_ISA, PRIMARY\_HEAP\_SIZE

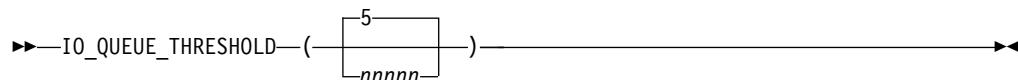
---

## **IO\_QUEUE\_THRESHOLD**

Use the IO\_QUEUE\_THRESHOLD statement to define the number of log requests to hold in the I/O request queue.

The IO\_QUEUE\_THRESHOLD statement has the following syntax:

### **IO\_QUEUE\_THRESHOLD**



*where:*

**5|nnnnn**

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 statement that can be reloaded. If you reload RODM, RODM uses the keyword values contained in the customization file that 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**

Use the LOG\_LEVEL statement 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 LOG\_LEVEL statement has the following syntax:

### **LOG\_LEVEL**



*where:*

8|nnn

Specifies the API log level. You can specify a value 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 statement that you can reload. 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. A shortage of auxiliary storage can occur in a high-stress environment.

**Related Statements:** MLOG\_LEVEL

## **MAX\_CHUNK**

Use the MAX\_CHUNK statement to define the maximum number of free chunks of storage kept for RODM methods. Also use this statement to control the amount of free short-lived storage allowed to accumulate before it is freed.

The MAX\_CHUNK statement has the following syntax:

### **MAX\_CHUNK**



*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 statement that you can reload. 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**

Use the MAX\_SEGMENT\_NUM statement to define the maximum number of available segments allowed in the translation window. The size of each segment is 16 MB.

The MAX\_SEGMENT\_NUM statement has the following syntax:

#### **MAX\_SEGMENT\_NUM**



*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 statement that you can reload. 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**

### **Purpose**

Use the MAX\_WINDOW\_NUM statement to define the maximum number of available windows in the data spaces. The size of each window is 16 MB.

### **Syntax**

The MAX\_WINDOW\_NUM statement has the following syntax:

## **MAX\_WINDOW\_NUM**



*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**

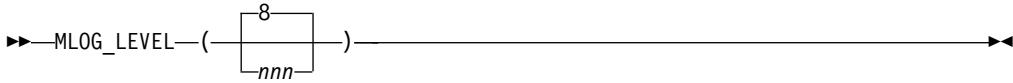
### **Purpose**

Use the MLOG\_LEVEL statement 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.

### **Syntax**

The MLOG\_LEVEL statement has the following syntax:

#### **MLOG\_LEVEL**



*where:*

**8|nnn**

Specifies the MAPI log level. You can specify a value in the range of 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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

## Related statements

LOG\_LEVEL

---

## MTRACE\_TYPE

### Purpose

Use the MTRACE\_TYPE statement to define the bitmap that specifies enabling and disabling of method tracing.

### Syntax

The MTRACE\_TYPE statement has the following syntax:

#### MTRACE\_TYPE

```
►►—MTRACE_TYPE—(—X—'—000000FC—'—X—'—nnnnnnnn—'—)—————►►
```

*where:*

**X'000000FC' | X'nnnnnnnn.'**

Specifies the bitmap. 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 bitmap also specifies the types of methods to be traced for a user in the following ways:

#### Value   Method Traced

**X'00000001'**  
Method entry

**X'00000002'**  
Method exit

**X'00000004'**  
QUERY method

**X'00000008'**  
CHANGE method

**X'00000010'**  
NOTIFY method

**X'00000020'**  
NAMED method

**X'00000040'**  
OI method

**X'00000080'**  
Object-deletion method

- 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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

---

## PLI\_ISA

### Purpose

Use the PLI\_ISA statement to define the size of the PL/I internal storage area (ISA), in bytes.

### Syntax

The PLI\_ISA statement has the following syntax:

#### PLI\_ISA



where:

40K | 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, MVS storage requests increase. 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 runtime 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, in addition to 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 statement that you can reload. 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

### Purpose

Use the PRIMARY\_HEAP\_SIZE statement to define the size, in bytes, of the primary PL/I heap storage for each user thread.

## Syntax

The PRIMARY\_HEAP\_SIZE statement has the following syntax:

### PRIMARY\_HEAP\_SIZE



*where:*

**64K | 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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

## Related statements

PLI\_ISA, EXTEND\_HEAP\_SIZE

---

## QUIES\_WAIT\_TASK\_TIME

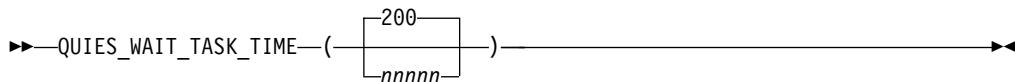
### Purpose

The QUIES\_WAIT\_TASK\_TIME statement defines 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.

## Syntax

The QUIES\_WAIT\_TASK\_TIME statement has the following syntax:

### QUIES\_WAIT\_TASK\_TIME



*where:*

**200 | nnnnn**

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 statement that you can reload. 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

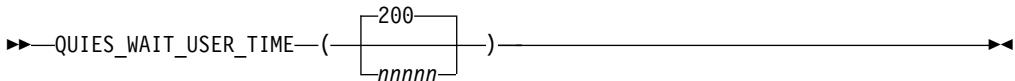
### Purpose

Use the QUIES\_WAIT\_USER\_TIME statement 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.

### Syntax

The QUIES\_WAIT\_USER\_TIME statement has the following syntax:

#### QUIES\_WAIT\_USER\_TIME



```
►► QUIES_WAIT_USER_TIME ([200 | nnnnn]) ──────────►
```

where:

**200 | nnnnn**

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

### Purpose

Use the SEC\_CLASS statement to define the security class name used by your system authorization facility (SAF) product. Code this statement to correspond to your SAF product.

## Syntax

The SEC\_CLASS statement has the following syntax:

### 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.
- 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 RODMMGR 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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

## Usage notes

SEC\_RNAME

---

## SEC\_RNAME

### Purpose

The SEC\_RNAME statement specifies a prefix for the resource names that are used to check the six RODM authority levels, as shown in Table 17 on page 559, in the SAF product. The resource name is the prefix that you specify with a digit (1–6) appended. For example, if the SEC\_RNAME statement is specified as SEC\_RNAME(RODM), the following resource names are used in the SAF product:

```
RODM1
RODM2
RODM3
RODM4
RODM5
RODM6
```

### Syntax

The SEC\_RNAME statement has the following syntax:

#### **SEC\_RNAME**

►►SEC\_RNAME—(*resname*)—►►

*where:*

*resname*

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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

*Table 17. RODM Access Authority Levels*

| Authority Level | Capabilities                                                                   |
|-----------------|--------------------------------------------------------------------------------|
| 1               | Connecting                                                                     |
| 2               | Query and list of functions (queries only)                                     |
| 3               | Action and list of functions (queries or actions) including triggering methods |
| 4               | Checkpointing                                                                  |
| 5               | Administrative functions                                                       |
| 6               | Stopping                                                                       |

## Related statements

SEC\_CLASS

## SEGMENT\_POCKETS

### Purpose

Use the SEGMENT\_POCKETS statement 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.

### Syntax

The SEGMENT\_POCKETS statement has the following syntax:

#### SEGMENT\_POCKETS



*where:*

1|nn

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 statement that you can reload. 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**

### **Purpose**

Use the SSB\_CHAIN statement to define the number of same-name system status blocks (SSBs) that can concurrently exist in the system.

### **Syntax**

The SSB\_CHAIN statement has the following syntax:

#### **SSB\_CHAIN**



*where:*

3|nnnnn

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 statement that you can reload. 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**

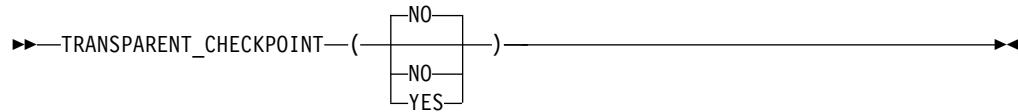
### **Purpose**

Use the TRANSPARENT\_CHECKPOINT statement to specify whether to queue up the UAPI and MAPI requests while RODM is checkpointing.

### **Syntax**

The TRANSPARENT\_CHECKPOINT statement has the following syntax:

#### **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 statement that you can reload. 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

### Purpose

Use the WAIT\_ALLOC\_TIME statement to define the waiting time, in increments of 0.01 seconds, for allocating window segments before RODM interprets the attempt as a failure.

### Syntax

The WAIT\_ALLOC\_TIME statement has the following syntax:

#### 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

### Purpose

Use the WINDOW\_CHKPT\_TIME statement 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.

### Syntax

The WINDOW\_CHKPT\_TIME statement has the following syntax:

#### WINDOW\_CHKPT\_TIME

```
►►WINDOW_CHKPT_TIME-(1500)nnnn)►►
```

where:

1500|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

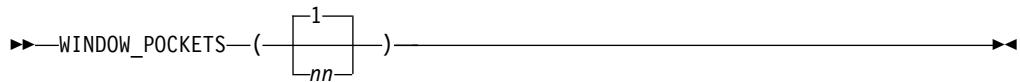
### Purpose

Use the WINDOW\_POCKETS statement 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.

### Syntax

The WINDOW\_POCKETS statement has the following syntax:

## **WINDOW\_POCKETS**



*where:*

1|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 statement that you can reload. If you reload the customization file, RODM uses the original keyword values specified when you started RODM.

### **Related statements**

MAX\_WINDOW\_NUM, SEGMENT\_POCKETS



---

## Chapter 9. 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

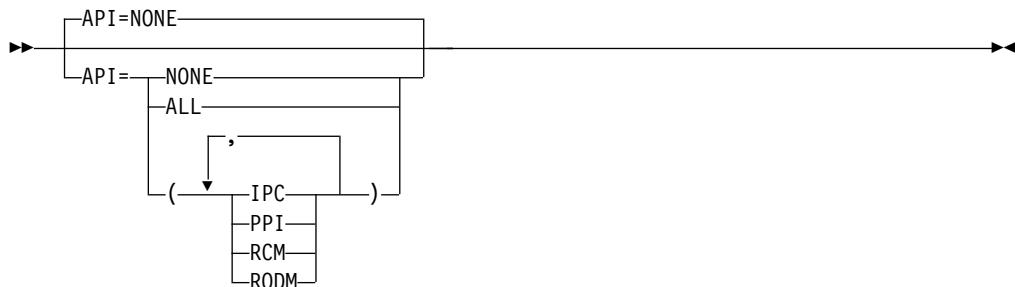
#### Purpose

The API statement specifies the application programming interface to be traced if a Graphical Monitor Facility host subsystem (GMFHS) trace is started.

#### Syntax

The API statement has the following syntax:

##### 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 values:

- 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 is specified with any keyword other than NONE, ALL is used as the initialization value.

- If NONE is specified with any keyword other than ALL, NONE is used as the initialization value.
- If ALL and NONE both are specified 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 run tracing unless necessary for debugging, and run tracing under low volume, if possible.

## Related statements

LEVEL, PRINTPDU38, STORAGE, TASK, TRACE, TYPE

## CHECKPOINT

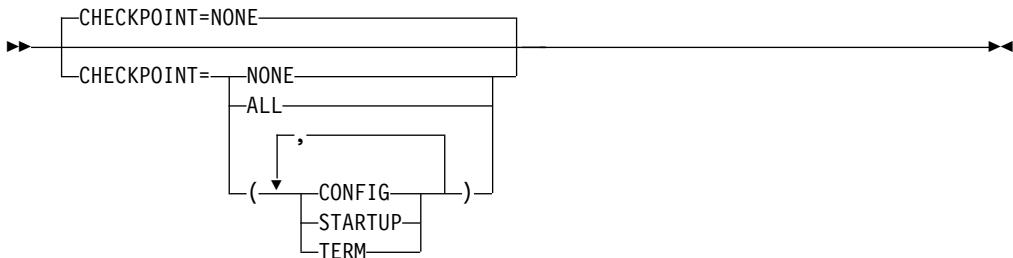
### Purpose

The CHECKPOINT statement specifies when GMFHS requests RODM checkpoints.

### Syntax

The CHECKPOINT statement has the following syntax:

#### 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 is specified 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

### Purpose

The DOMAIN statement identifies the domain to which this GMFHS component belongs. This is a required statement.

### Syntax

The DOMAIN statement has the following syntax:

#### DOMAIN

►—DOMAIN—=—*domainid*—►

*where:*

*domainid*

Indicates the ID of the GMFHS component (a maximum of five alphanumeric characters).

### Usage notes

DOMAIN is used to provide a unique identification of the GMFHS component. Set DOMAIN to the same value specified in the NetView address space. GMFHS does not start if this parameter is not specified.

---

## GMTOFFSET

### Purpose

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).

### Syntax

The GMTOFFSET statement has the following syntax:

#### GMTOFFSET

►►—GMTOFFSET—=—*shmm*—►►

*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.
- 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

### Purpose

The JAPANESE statement specifies whether GMFHS uses Japanese text for any displayed text that it provides to a NetView management console console.

### Syntax

The JAPANESE statement has the following syntax:

**JAPANESE**

►►—

|              |
|--------------|
| JAPANESE=OFF |
| JAPANESE=ON  |

—►►

*where:*

**OFF**

Disables translation to Japanese. This is the default.

**ON** Enables translation to Japanese.

---

## LCON-AGG-BUNDLE-INTERVAL

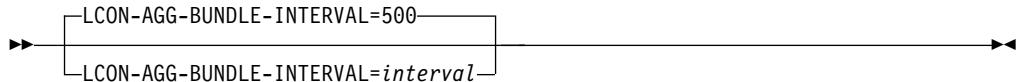
### Purpose

The LCON-AGG-BUNDLE-INTERVAL statement specifies the time interval between calls of DUIFCASB to update aggregate object counts.

### Syntax

The LCON-AGG-BUNDLE-INTERVAL statement has the following syntax:

## LCON-AGG-BUNDLE-INTERVAL



*where:*

**500 | interval**

Indicates a value, in hundredths of a second, in the range 10 - 864000000000. The default value specified in the sample is 500.

### Usage notes

Because this is the only routine that changes Aggregate object counts, you do not need to explicitly lock lists of objects to ensure that asynchronous changes are made to an Aggregate currently being processed.

---

## LCON-AGGRST-REQUIRED

### Purpose

Use the LCON-AGGRST-REQUIRED statement 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.

### Syntax

The LCON-AGGRST-REQUIRED statement has the following syntax:

#### LCON-AGGRST-REQUIRED



*where:*

**NO** Specifies not to run the method. This is the default.

**YES**

Specifies to run the DUFFAWS aggregation warm-start method at startup and after a CONFIG NETWORK command.

---

## LCON-AIP-RESET-INTERVAL

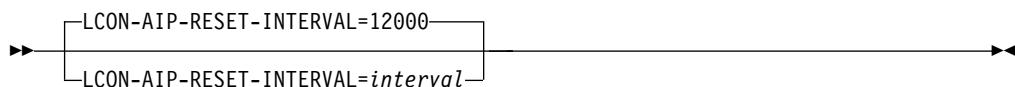
### Purpose

The LCON-AIP-RESET-INTERVAL statement specifies the time interval between calls 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.

## Syntax

The LCON-AIP-RESET-INTERVAL statement has the following syntax:

### LCON-AIP-RESET-INTERVAL



where:

**12000 | interval**

Is a value, in hundredths of a second, in the range 0 - 864000000000. The minimum value of zero (0) indicates that GMFHS does not monitor the AIP bit to determine if it needs to be reset. The maximum value of 864000000 is equivalent to 100 days. The default value specified in the sample is 12000 (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 calls of method DUILFRAIP which determines if the AIP bit is reset.

---

## LCON-ALERT-CMD-TIMEOUT

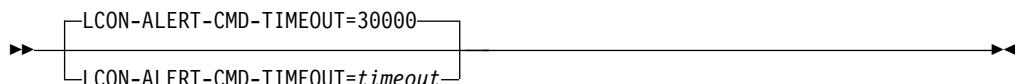
### Purpose

The LCON-ALERT-CMD-TIMEOUT statement specifies the number of seconds, multiplied by 0.01, Dbserver waits before stopping the alert history request of an NetView management console client waiting for reply data. The timeout while waiting for data that can occur at the NetView management console client is unaffected by this setting.

## Syntax

The LCON-ALERT-CMD-TIMEOUT statement has the following syntax:

### LCON-ALERT-CMD-TIMEOUT



where:

**30000 | timeout**

Indicates a value, in hundredths of a second, in the range 100 - 360000. The default value specified in the sample is 30000.

---

## LCON-ASSOCIATE-NULL-NODE-WITH-LINK

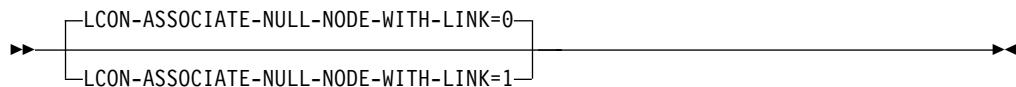
### Purpose

The LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement specifies how null nodes are to 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.

### Syntax

The LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement has the following syntax:

#### 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 can result in potential layout problems for null nodes in customized views. However, changing the default value can 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 are lost.

### Usage notes

Changing the default value can result in problems migrating customized views.

---

## LCON-EVCHANGE-BUFFER-INTERVAL

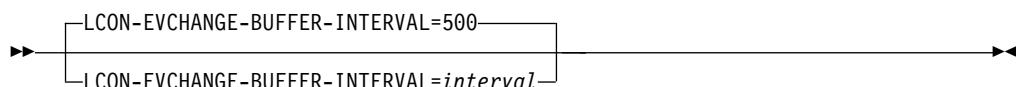
### Purpose

The LCON-EVCHANGE-BUFFER-INTERVAL statement specifies the time interval between calls of the resource traits change method for exception views.

### Syntax

The LCON-EVCHANGE-BUFFER-INTERVAL statement has the following syntax:

#### LCON-EVCHANGE-BUFFER-INTERVAL



where:

**500|interval**

Indicates a value, in hundredths of a second, in the range 10 - 864000000000. The default value specified in the sample is 500.

### Usage notes

When the interval expires, the resource traits change method is triggered to bundle view updates. The view updates consist of data relevant to Display or User status changes or ExceptionViewFilter or ExceptionViewList fields.

---

## LCON-HEX-SUBVECTOR-DISPLAY

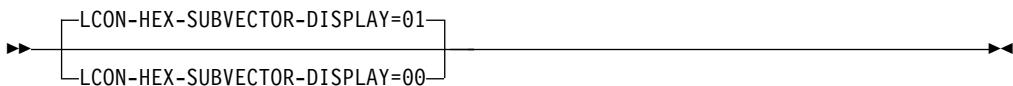
### Purpose

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.

### Syntax

The LCON-HEX-SUBVECTOR-DISPLAY statement has the following syntax:

#### LCON-HEX-SUBVECTOR-DISPLAY



where:

- 01** 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** Indicates that a display of the hexadecimal dump of subvectors is not to be generated. You can also specify this value as 0.

### Usage notes

- If you do not code this statement in DUINIT, 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-VIEWS

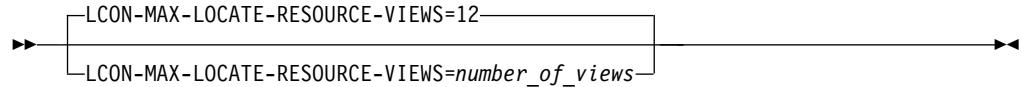
### Purpose

The LCON-MAX-LOCATE-RESOURCE-VIEWS statement specifies the maximum number of views to be returned on a Locate Resource request.

### Syntax

The LCON-MAX-LOCATE-RESOURCE-VIEWS statement has the following syntax:

#### LCON-MAX-LOCATE-RESOURCE-VIEWS



*where:*

**12|number\_of\_views**

Specifies the maximum number of views to be returned on a Locate Resource request. Valid values are in the range 1 - 32767. The default value is 12.

### Usage notes

Increasing the *number\_of\_views* value can cause timeouts- on the workstation.

## LCON-MAX-QUEUE-DBSERVER

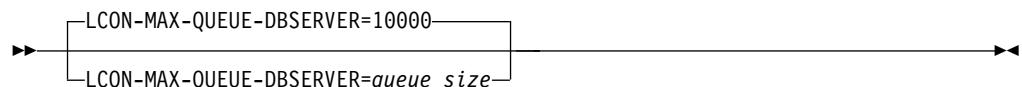
### Purpose

The LCON-MAX-QUEUE-DBSERVER statement specifies the maximum number of messages that can be held on the Dbserver task message queue.

### Syntax

The LCON-MAX-QUEUE-DBSERVER statement has the following syntax:

#### LCON-MAX-QUEUE-DBSERVER



*where:*

**10000|queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-MAINTASK, LCON-MAX-QUEUE-NETCMD,  
 LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
 LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
 LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-EVENTMGR

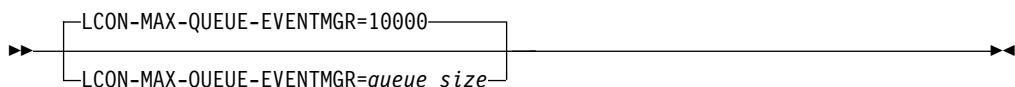
### Purpose

The LCON-MAX-QUEUE-EVENTMGR statement specifies the maximum number of messages that can be held on the EVENTMGR task message queue.

### Syntax

The LCON-MAX-QUEUE-EVENTMGR statement has the following syntax:

#### **LCON-MAX-QUEUE-EVENTMGR**



*where:*

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of sustained high alert traffic because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-IPC, LCON-MAX-QUEUE-IRMGR,  
LCON-MAX-QUEUE-MAINTASK, LCON-MAX-QUEUE-NETCMD,  
LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-IPC

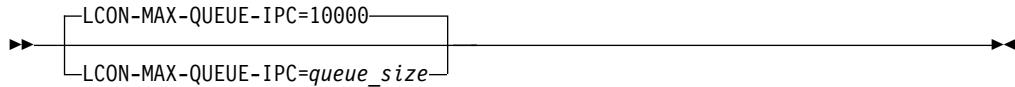
### Purpose

The LCON-MAX-QUEUE-IPC statement specifies the maximum number of messages that can be held on the IPC task message queue.

### Syntax

The LCON-MAX-QUEUE-IPC statement has the following syntax:

## LCON-MAX-QUEUE-IPC



where:

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IRMGR,  
LCON-MAX-QUEUE-MAINTASK, LCON-MAX-QUEUE-NETCMD,  
LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-IRMGR

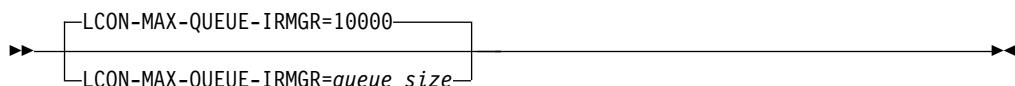
### Purpose

The LCON-MAX-QUEUE-IRMGR statement specifies the maximum number of messages that can be held on the IRMGR task message queue.

### Syntax

The LCON-MAX-QUEUE-IRMGR statement has the following syntax:

## LCON-MAX-QUEUE-IRMGR



where:

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
 LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
 LCON-MAX-QUEUE-MAINTASK, LCON-MAX-QUEUE-NETCMD,  
 LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
 LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
 LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

## LCON-MAX-QUEUE-MAINTASK

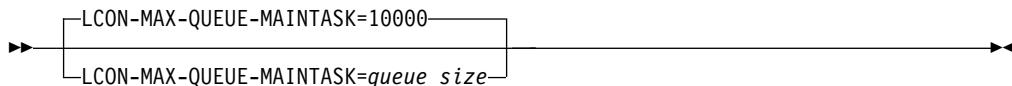
### Purpose

The LCON-MAX-QUEUE-MAINTASK statement specifies the maximum number of messages that can be held on the MAINTASK task message queue.

### Syntax

The LCON-MAX-QUEUE-MAINTASK statement has the following syntax:

#### **LCON-MAX-QUEUE-MAINTASK**



*where:*

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
 LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
 LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-NETCMD,  
 LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
 LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
 LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-NETCMD

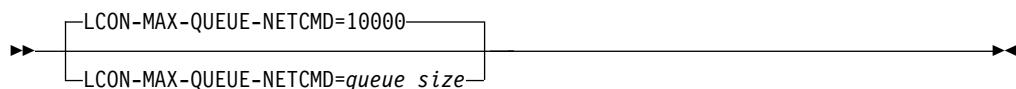
### Purpose

The LCON-MAX-QUEUE-NETCMD statement specifies the maximum number of messages that can be held on the NETCMD task message queue.

### Syntax

The LCON-MAX-QUEUE-NETCMD statement has the following syntax:

#### LCON-MAX-QUEUE-NETCMD



where:

**10000|queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
LCON-MAX-QUEUE-NETCON, LCON-MAX-QUEUE-OPERIF,  
LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-NETCON

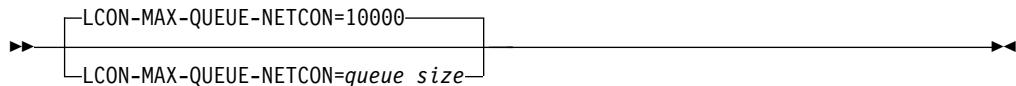
### Purpose

The LCON-MAX-QUEUE-NETCON statement specifies the maximum number of messages that can be held on the NETCON task message queue.

### Syntax

The LCON-MAX-QUEUE-NETCON statement has the following syntax:

## LCON-MAX-QUEUE-NETCON



where:

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-OPERIF,  
LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-OPERIF

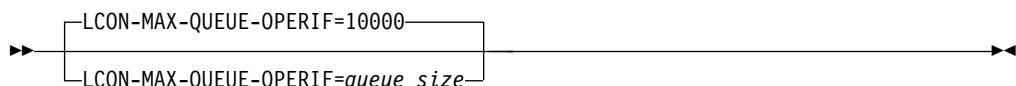
### Purpose

The LCON-MAX-QUEUE-OPERIF statement specifies the maximum number of messages that can be held on the OPERIF task message queue.

### Syntax

The LCON-MAX-QUEUE-OPERIF statement has the following syntax:

## LCON-MAX-QUEUE-OPERIF



where:

**10000 | queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disk constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
 LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
 LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
 LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-NETCON,  
 LCON-MAX-QUEUE-RCMGR, LCON-MAX-QUEUE-RTMGR,  
 LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

## LCON-MAX-QUEUE-RCMGR

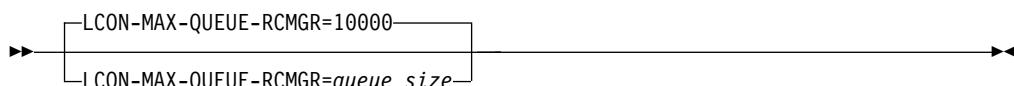
### Purpose

The LCON-MAX-QUEUE-RCMGR statement specifies the maximum number of messages that can be held on the RCMGR task message queue.

### Syntax

The LCON-MAX-QUEUE-RCMGR statement has the following syntax:

#### **LCON-MAX-QUEUE-RCMGR**



*where:*

**10000|queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disk constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
 LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
 LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
 LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-NETCON,  
 LCON-MAX-QUEUE-OPERIF, LCON-MAX-QUEUE-RTMGR,  
 LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-RTMGR

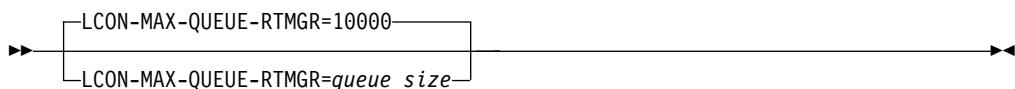
### Purpose

The LCON-MAX-QUEUE-RTMGR statement specifies the maximum queue size for the Resource Trait subtask.

### Syntax

The LCON-MAX-QUEUE-RTMGR statement has the following syntax:

#### LCON-MAX-QUEUE-RTMGR



where:

**10000 | queue\_size**

Indicates a value in the range 100 - 10000. The default value specified in the sample is 10000.

### Usage notes

If you specify a value greater than 10000, message DUI4074E is issued and the default of 10000 is used.

### Related statements

LCON-MAX-QUEUE-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-NETCON,  
LCON-MAX-QUEUE-OPERIF, LCON-MAX-QUEUE-RCMGR,  
LCON-MAX-QUEUE-VIEWMGR, LCON-MAX-QUEUE-VSTATMGR

---

## LCON-MAX-QUEUE-VIEWMGR

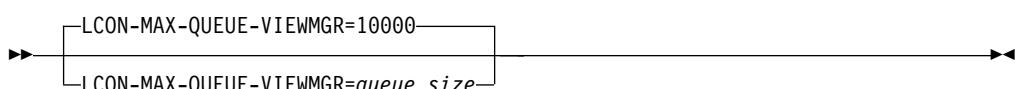
### Purpose

The LCON-MAX-QUEUE-VIEWMGR statement specifies the maximum number of messages that can be held on the VIEWMGR task message queue.

### Syntax

The LCON-MAX-QUEUE-VIEWMGR statement has the following syntax:

#### LCON-MAX-QUEUE-VIEWMGR



where:

### **10000|queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-NETCON,  
LCON-MAX-QUEUE-OPERIF, LCON-MAX-QUEUE-RCMGR,  
LCON-MAX-QUEUE-RTMGR, LCON-MAX-QUEUE-VSTATMGR

---

## **LCON-MAX-QUEUE-VSTATMGR**

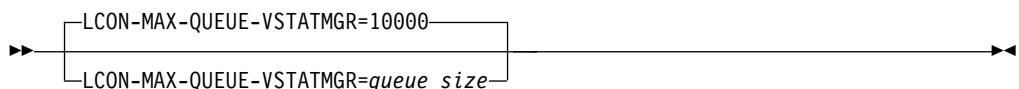
### **Purpose**

The LCON-MAX-QUEUE-VSTATMGR statement specifies the maximum number of messages that can be held on the VSTATMGR task message queue.

### **Syntax**

The LCON-MAX-QUEUE-VSTATMGR statement has the following syntax:

#### **LCON-MAX-QUEUE-VSTATMGR**



*where:*

### **10000|queue\_size**

Indicates a value in the range 100 - 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 a possibility of your disks constantly being accessed because of asynchronous events exists. 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-DBSERVER,  
LCON-MAX-QUEUE-EVENTMGR, LCON-MAX-QUEUE-IPC,  
LCON-MAX-QUEUE-IRMGR, LCON-MAX-QUEUE-MAINTASK,  
LCON-MAX-QUEUE-NETCMD, LCON-MAX-QUEUE-NETCON,  
LCON-MAX-QUEUE-OPERIF, LCON-MAX-QUEUE-RCMGR,  
LCON-MAX-QUEUE-RTMGR, LCON-MAX-QUEUE-VIEWMGR

---

## LCON-NCC-RETRY-LIMIT

### Purpose

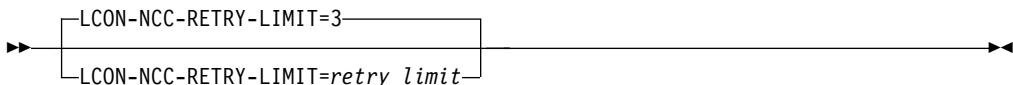
The LCON-NCC-RETRY-LIMIT statement specifies the number of times GMFHS attempts to resend a status solicitation network command to a native element manager.

**Note:** LCON-NCC-RETRY-LIMIT does not apply to commands issued from the NetView management console

### Syntax

The LCON-NCC-RETRY-LIMIT statement has the following syntax:

#### LCON-NCC-RETRY-LIMIT



where:

3|retry\_limit

Indicates a value in the range 0 - 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 cannot be processed because of a temporary condition (for example, Resource Busy or Currently Not Available), use LCON-NCC-RETRY-LIMIT 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

### Purpose

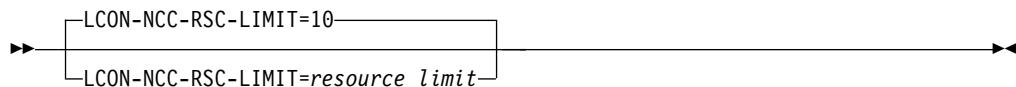
The LCON-NCC-RSC-LIMIT statement specifies the maximum number of resources that the network command manager task can name in a status solicitation network command (for example, Display Status) for initial status for those native element managers that use the DOMP010 protocol. This group of resources 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.

**Note:** LCON-NCC-RCS-LIMIT does not apply to commands issued from the NetView management console workstation.

### Syntax

The LCON-NCC-RSC-LIMIT statement has the following syntax:

#### LCON-NCC-RSC-LIMIT



where:

10|resource\_limit

Indicates a value in the range 1 - 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

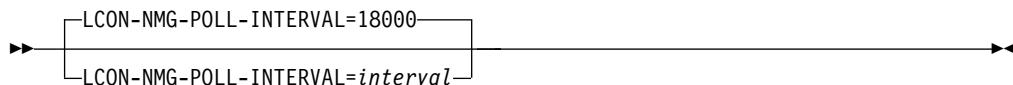
### Purpose

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.

## Syntax

The LCON-NMG-POLL-INTERVAL statement has the following syntax:

### LCON-NMG-POLL-INTERVAL



*where:*

**18000 | interval**

Indicates a value, in hundredths of a second, in the range 100 - 360000. The default value is 18000 (3 minutes).

### Usage notes

- Ensure that this value allows 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

### Purpose

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 is to 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.

## Syntax

The LCON-OPERATOR-CMD-AUDIT statement has the following syntax:

### 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 notes

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

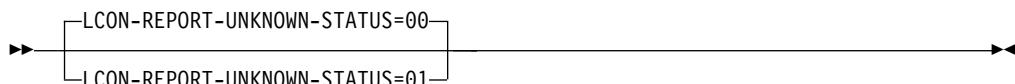
### Purpose

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.

### Syntax

The LCON-REPORT-UNKNOWN-STATUS statement has the following syntax:

#### LCON-REPORT-UNKNOWN-STATUS



where:

- 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 notes

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

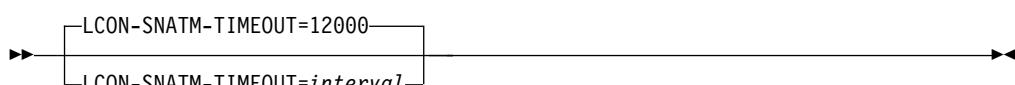
### Purpose

The LCON-SNATM-TIMEOUT statement indicates how long (in hundredths of seconds) the Graphic Monitor Facility host subsystem (GMFHS) is to wait before timing out, for the SNA Topology Manager response to a locate resource request.

### Syntax

The LCON-SNATM-TIMEOUT statement has the following syntax:

#### LCON-SNATM-TIMEOUT



*where:*

**12000|interval**

Indicates a value, in hundredths of a second, in the range 100 - 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

### Purpose

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.

### Syntax

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 LCON-STATUS-DELAY-MAX statement has the following syntax:

### LCON-STATUS-DELAY-MAX



*where:*

**10|count**

Specifies the maximum number of LCON-STATUS-DELAY-TIME intervals. This value is in the range 0 - 50. The default value is 10.

### Usage notes

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.

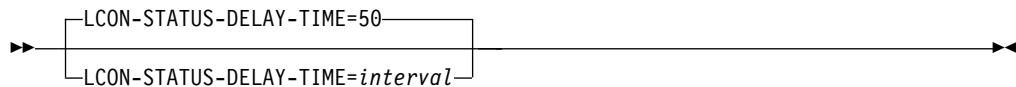
### Purpose

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.

### Syntax

The LCON-STATUS-DELAY-TIME statement has the following syntax:

#### 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 value is in the range 10 - 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

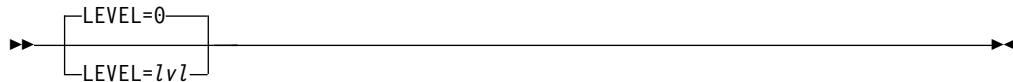
### Purpose

The LEVEL statement indicates the minimum level of program trace statement that is included in each trace entry.

### Syntax

The LEVEL statement has the following syntax:

## **LEVEL**



*where:*

**0|lv1**

Specifies the minimum level of program trace statement that is included. The range is 0 - 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 run tracing unless necessary for debugging, and run tracing under low volume, if possible.

### **Related statements**

API, PRINTPDU38, STORAGE, TASK, TRACE, TYPE

## **PRINTPDU38**

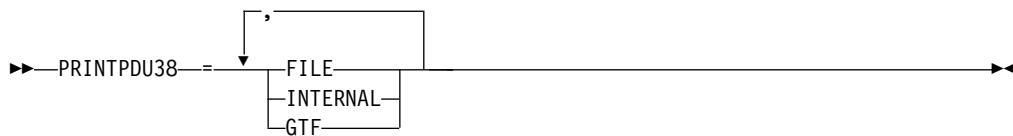
### **Purpose**

The PRINTPDU38 statement specifies where to log system error synopsis (PDU38) information and trace data.

### **Syntax**

The PRINTPDU38 statement has the following syntax:

#### **PRINTPDU38**



*where:*

#### **FILE**

Indicates that the data is written to the GMFHS output data sets for the appropriate GMFHS subcomponents. The output data sets are defined by GMFHS using the following DD statements in the GMFHS startup procedure:

|                 |                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>CNMC</b>     | Network command manager (NETCMD)                                                                                                  |
| <b>CNMD</b>     | Database server (DBSERVER)                                                                                                        |
| <b>CNME</b>     | Event manager (EVENTMGR)                                                                                                          |
| <b>CNMF</b>     | Network configuration manager (NETCON)                                                                                            |
| <b>CNMI</b>     | Interprocessor communication (IPC)                                                                                                |
| <b>CNMM</b>     | GMFHS main (control) task (MAINTASK)                                                                                              |
| <b>CNMO</b>     | Operator interface manager (OPERIF)                                                                                               |
| <b>CNMP</b>     | IPC-RODM event manager (IRMGR)                                                                                                    |
| <b>CNMR</b>     | Resource traits manager (RTMGR)                                                                                                   |
| <b>CNMS</b>     | View status manager (VSTATMGR)                                                                                                    |
| <b>CNMV</b>     | View manager (VIEWMGR)                                                                                                            |
| <b>INTERNAL</b> | Indicates that the data is written to the GMFHS internal trace log.                                                               |
| <b>GTF</b>      | Indicates that the data is written to GTF.<br><br>The event identifier (EID) used for the TRACE records written to GTF is X'5E2'. |

### Usage notes

- If multiple locations are listed, the data is written to each location specified.
- If you specify a value that is not valid, message DUI4076E is issued, and the default of 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 run tracing unless necessary for debugging, and run 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 no other output logs are available to log PDU38 data, the internal trace log is temporarily activated. When one of the other output logs becomes available, the internal trace log is deactivated.

### Related statements

API, LEVEL, STORAGE, TASK, TRACE, TRACEBYTES, TRACEPAGES, TYPE

---

## RODMID

### Purpose

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.

### Syntax

The RODMID statement has the following syntax:

#### RODMID

►►RODMID---*rodmid*-----►►

*where:*

*rodmid*

Indicates the ID of the RODM user (a maximum of eight 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

### Purpose

The RODMNAME statement specifies the name of the RODM that GMFHS uses. This is a required statement.

### Syntax

The RODMNAME statement has the following syntax:

#### RODMNAME

►►RODMNAME---*rodname*-----►►

*where:*

*rodname*

Indicates the name of the RODM (a maximum of 8 characters).

## Usage notes

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

## Purpose

The STORAGE statement specifies whether get and free storage tracing is on or off.

## Syntax

The STORAGE statement has the following syntax:

### 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 run tracing unless necessary for debugging, and run tracing under low volume, if possible.

## Related statements

API, LEVEL, PRINTPDU38, TASK, TRACE, TYPE

---

# TASK

## Purpose

The TASK statement indicates the NetView GMFHS tasks to be traced.

## Syntax

The TASK statement has the following syntax:

## **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 run tracing unless necessary for debugging, and run tracing under low volume, if possible.

### **Related statements**

API, LEVEL, PRINTPDU38, STORAGE, TRACE, TYPE

---

## **TRACE**

### **Purpose**

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.

**Attention:** Logging trace information to the VSAM database can adversely affect performance in logging and in accessing alert information.

## Syntax

The TRACE statement has the following syntax:

### 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 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 run tracing unless necessary for debugging, and run tracing under low volume, if possible.

## Related statements

API, LEVEL, PRINTPDU38, STORAGE, TASK, TYPE

---

## TRACEBYTES

### Purpose

Use the TRACEBYTES statement to specify how many bytes of each trace record are written to the internal trace log. This is an optional statement.

### Syntax

The TRACEBYTES statement has the following syntax:

### TRACEBYTES



*where:*

**0|bytes**

Indicates the number of bytes from each trace record that is written to the internal trace log. You can specify a value in the range 32 - 32656. Zero (0) is also a valid value. The default value is 0.

### Usage notes

Specify a number of TRACEBYTES that fits evenly into a 4 KB page.

### Related statements

TRACE, TRACEPAGES

---

## TRACEPAGES

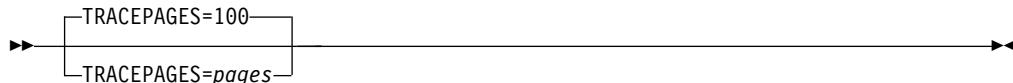
### Purpose

Use the TRACEPAGES statement to specify the number of pages of storage to allocate for the internal trace log for GMFHS. This is an optional statement.

### Syntax

The TRACEPAGES statement has the following syntax:

#### 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 - 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

### Purpose

The TYPE statement instructs the tasks to include trace entries for specific interface and message types.

## Syntax

The TYPE statement has the following syntax:

### TYPE

►—TYPE—=—*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 run tracing unless necessary for debugging, and run tracing under low volume.

## Related statements

API, LEVEL, PRINTPDU38, STORAGE, TASK, TRACE



---

## **Appendix. Common Global Variables for the Canzlog Archiving Function**

When the NetView program reads the Canzlog archive-related statements that begin with ARCHIVE. in the NetView style sheet, such as sample CNMSTYLE and its included members, the program stores information in common global variables. The DSLOGMT task processes the common global variables to determine how a Canzlog archive is to be used.

The names of common global variables related to Canzlog archiving and the formats of their values are shown here to provide information that may be used to manage archived Canzlog data.

### **CNMSTYLE.CZLARCH.HLQ**

The value is the high-level qualifier that is specified for the ARCHIVE.HLQ statement in the NetView style sheet. If the ARCHIVE.HLQ statement is omitted from the NetView style sheet or ARCHIVE.HLQ = \*NONE\* is coded in the NetView style sheet, the common global variable will not be defined.

### **CNMSTYLE.CZLARCH.WRITE**

The value is 1 if ARCHIVE.WRITE = Yes is coded in the NetView style sheet and a valid high-level qualifier is coded for the ARCHIVE.HLQ statement. Otherwise, the value is 0.

The NetView program sets the value to 0, instead of 1, when ARCHIVE.WRITE = Yes is coded in the NetView style sheet and one of the following conditions is true:

- The ARCHIVE.HLQ statement is not specified, specifies \*NONE\*, or has an invalid value.
- A valid value other than \*NONE\* is specified for the ARCHIVE.HLQ statement and the ARCHIVE.ACCESSDELAY statement specifies a valid time > 00:00:00.

After the time interval specified for the ARCHIVE.ACCESSDELAY statement expires, the NetView program sets the value to 1.

A value of 1 for an instance of the NetView program does not necessarily mean that instance is writing Canzlog archive data sets. If multiple instances of the NetView program are running on one z/OS image and each instance specifies ARCHIVE.WRITE = Yes and a valid qualifier for ARCHIVE.HLQ in its NetView style sheet, only the instance of the NetView program that successfully reserves the right to write Canzlog archive data sets can write the data sets.

### **CNMSTYLE.CZLARCH.INDEX.SMS**

The value is a 24-character string that contains any SMS classes that are defined for Canzlog archive index data sets in the NetView style sheet via the ARCHIVE.INDEX.DATACLAS, ARCHIVE.INDEX.STORCLAS, and ARCHIVE.INDEX.MGMTCLAS statements. The format of this common global variable's value is as follows:

#### **Characters 1-8**

The value comes from the ARCHIVE.INDEX.DATACLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the

field. If no ARCHIVE.INDEX.DATACLAS statement is coded or ARCHIVE.INDEX.DATACLAS = \*NONE\* is coded, the 8 characters are all blanks.

#### **Characters 9-16**

The value comes from the ARCHIVE.INDEX.STORCLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.INDEX.STORCLAS statement is coded or ARCHIVE.INDEX.STORCLAS = \*NONE\* is coded, the 8 characters are all blanks.

#### **Characters 17-24**

The value comes from the ARCHIVE.INDEX.MGMTCLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.INDEX.MGMTCLAS statement is coded or ARCHIVE.INDEX.MGMTCLAS = \*NONE\* is coded, the 8 characters are all blanks.

### **CNMSTYLE.CZLARCH.INDEX.DSCHAR**

The value is a 22-character string that contains data set characteristics for Canzlog archive index data sets. The format of this common global variable's value is as follows:

#### **Characters 1-5**

The value comes from the ARCHIVE.INDEX.BLOCKSIZE statement. If no ARCHIVE.INDEX.BLOCKSIZE statement is coded, the value is the default block size for Canzlog archive index data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

#### **Characters 6-14**

The value is the space unit, such as BLOCKS, TRACKS, or CYLINDERS, which is indicated by the first value of the ARCHIVE.INDEX.SPACE statement. If no ARCHIVE.INDEX.SPACE statement is coded, the value is the default space units for Canzlog archive index data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

#### **Characters 15-18**

The value is the primary space allocation from the ARCHIVE.INDEX.SPACE statement. If no ARCHIVE.INDEX.SPACE statement is coded, the value is the default primary space allocation for Canzlog archive index data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

#### **Characters 19-22**

The value is the secondary space allocation from the ARCHIVE.INDEX.SPACE statement. If no ARCHIVE.INDEX.SPACE statement is coded, the value is the default secondary space allocation for Canzlog archive index data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

### **CNMSTYLE.CZLARCH.INDEX.VOLDATA**

The value is a 14-character string that contains volume and volume selection data for Canzlog archive index data sets. The format of this common global variable's value is as follows:

#### **Characters 1-8**

The value is the unit name (device type) from the ARCHIVE.INDEX.UNIT statement. If no ARCHIVE.INDEX.UNIT statement is coded, the value is the

default unit name for Canzlog archive index data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

#### **Characters 9-14**

The value is the volume serial number from the ARCHIVE.INDEX.VOLUME statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.INDEX.VOLUME statement is coded, the 6 characters are all blanks.

### **CNMSTYLE.CZLARCH.MSG.SMS**

The value is a 24-character string that contains any SMS classes that are defined for Canzlog archive message data sets in the NetView style sheet via the ARCHIVE.MESSAGE.DATACLAS, ARCHIVE.MESSAGE.STORCLAS, and ARCHIVE.MESSAGE.MGMTCLAS statements. The format of this common global variable's value is as follows:

#### **Characters 1-8**

The value comes from the ARCHIVE.MESSAGE.DATACLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.DATACLAS statement is coded or ARCHIVE.MESSAGE.DATACLAS = \*NONE\* is coded, the 8 characters are all blanks.

#### **Characters 9-16**

The value comes from the ARCHIVE.MESSAGE.STORCLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.STORCLAS statement is coded or ARCHIVE.MESSAGE.STORCLAS = \*NONE\* is coded, the 8 characters are all blanks.

#### **Characters 17-24**

The value comes from the ARCHIVE.MESSAGE.MGMTCLAS statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.MGMTCLAS statement is coded or ARCHIVE.MESSAGE.MGMTCLAS = \*NONE\* is coded, the 8 characters are all blanks.

### **CNMSTYLE.CZLARCH.MSG.DSCHAR**

The value is a 22-character string that contains data set characteristics for Canzlog archive message data sets. The format of this common global variable's value is as follows:

#### **Characters 1-5**

The value comes from the ARCHIVE.MESSAGE.BLOCKSIZE statement. If no ARCHIVE.MESSAGE.BLOCKSIZE statement is coded, the value is the default block size for Canzlog archive message data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

#### **Characters 6-14**

The value is the space unit, such as BLOCKS, TRACKS, or CYLINDERS, which is indicated by the first value of the ARCHIVE.MESSAGE.SPACE statement. If no ARCHIVE.MESSAGE.SPACE statement is coded, the value is the default space units for Canzlog archive message data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

### **Characters 15-18**

The value is the primary space allocation from the ARCHIVE.MESSAGE.SPACE statement. If no ARCHIVE.MESSAGE.SPACE statement is coded, the value is the default primary space allocation for Canzlog archive message data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

### **Characters 19-22**

The value is the secondary space allocation from the ARCHIVE.MESSAGE.SPACE statement. If no ARCHIVE.MESSAGE.SPACE statement is coded, the value is the default secondary space allocation for Canzlog archive message data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

## **CNMSTYLE.CZLARCH.MSG.VOLDATA**

The value is a 68-character string that contains volume and volume selection data for Canzlog archive message data sets. The format of this common global variable's value is as follows:

### **Characters 1-8**

The value is the unit name (device type) from the ARCHIVE.MESSAGE.UNIT statement. If no ARCHIVE.MESSAGE.UNIT statement is coded, the value is the default unit name for Canzlog archive message data sets. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field.

### **Characters 9-14**

The value is the first volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement is coded, the value is all blanks.

### **Characters 15-20**

The value is the second volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no second volume serial number is specified, the value is all blanks.

### **Characters 21-26**

The value is the third volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no third volume serial number is specified, the value is all blanks.

### **Characters 27-32**

The value is the fourth volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no fourth volume serial number is specified, the value is all blanks.

### **Characters 33-38**

The value is the fifth volume serial number from the

ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no fifth volume serial number is specified, the value is all blanks.

#### Characters 39-44

The value is the sixth volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no sixth volume serial number is specified, the value is all blanks.

#### Characters 45-50

The value is the seventh volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no seventh volume serial number is specified, the value is all blanks.

#### Characters 51-56

The value is the eighth volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no eighth volume serial number is specified, the value is all blanks.

#### Characters 57-62

The value is the ninth volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no ninth volume serial number is specified, the value is all blanks.

#### Characters 63-68

The value is the tenth volume serial number from the ARCHIVE.MESSAGE.VOLUMES statement. The value is left-aligned, and if necessary, is padded on the right with blanks to fill the field. If no ARCHIVE.MESSAGE.VOLUMES statement or an ARCHIVE.MESSAGE.VOLUMES statement is coded but no tenth volume serial number is specified, the value is all blanks.

The NetView program does not include information from the ARCHIVE.ACCESSDELAY and ARCHIVE.BROWSE.DATASPACES statements in the values of any common global variables.

**Note:** Do not change the values of these common global variables. Change the applicable Canzlog archive-related statements in the NetView style sheet and run the RESTYLE ARCHIVE command.



---

## Notices

This information was developed for products and services offered in the U.S.A.

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

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

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

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

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

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

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

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

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

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web

sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

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

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

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

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

---

## Programming Interfaces

This publication primarily documents information that is NOT intended to be used as Programming Interfaces of Tivoli NetView for z/OS. This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of Tivoli NetView for z/OS. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

Programming Interface information  
End of Programming Interface information

---

## Trademarks

IBM, the IBM logo, and ibm.com<sup>®</sup> are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe and Acrobat and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Other product and service names might be trademarks of IBM or other companies.

---

## Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM’s Privacy Policy at <http://www.ibm.com/privacy> and IBM’s Online Privacy Statement at <http://www.ibm.com/privacy/details> the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at <http://www.ibm.com/software/info/product-privacy>.



---

# Index

## Special characters

\*PRE keyword, KCLASS statement 340  
\*SEC keyword, KCLASS statement 340  
%INCLUDE statement 333

## A

A (Alert) statement 289

A01APPLS 1

A01SNA 1

AAUCNMTD 1

AAUPRMLP 1

ACBNAME keyword, VPDINIT statement 406

ACBpassword statement 7

access control block (ACB)

    primary data set 321

    secondary data set 321

ACCESS statement 291

accessibility xviii

action area, command facility panel 330

action messages, command facility panel 292

ACTION statement 292

ACTMON 409

AdapterCdsFile statement 502

AdapterFmtFile statement 503

additional status information, obtaining 388

addresses in the keyword list, F (Filter) statement 325

ADJNETV 414

administration, system 289

AFTER command 304

Aggregate object counts, changing 568

AIPTIMER bit 569

AL keyword, W (Wrap) statement 186

alarm operands 333

ALCACHE

    statement 168

alert

    cache size 168

    focal point 173, 314

    forwarding protocol 170

    log 172

    NPDA.ALERTLOG statement 172

    record 173

alert receiver 229

ALERTFWD statement 170

ALERTLOG statement 172

AlertRcvName statement 8

alias keyword, RMTALIAS statement 214

alias name

    assignment, logical unit 345

    translation tables 293

ALIASMEM statement 293

ALRTCFG statement 504

ALRTINFP statement 173

ALRTTCFG statement 505

ALT\_ALERT statement 174

AMELINIT statement 294

AMS 322

AON

    automation operator names 416

AON (*continued*)

    enabling 263

    IDS support 263

    SNA feature 263

    SNA X.25 support 263

    TCP/IP feature 263

AON Control File Entries, Defining 409

    ACTMON 409

    ADJNETV 414

    AUTOOPS 415

    CDLOG 419

    CPCPSESS (SNA) 421

    DDFGENERIC 422

    DDFGROUP 424

    ENVIRON AIP 424

    ENVIRON DDF 425

    ENVIRON EXIT 427

    ENVIRON RACF 429

    ENVIRON SETUP 431

    ENVIRON TIMEOUT 434

    EZLTLOG 435

    FORWARD FOCALPT 437

    FULLSESS 437

    GATEWAY 438

    INSTALLOPT 440

    IPCONN 442

    IPHOST 444

    IPINFC 446

    IPNAMESERV 448

    IPPORT 450

    IPROUTER 453

    IPTELNET 455

    IPTN3270 456

    LSTHRESH 459

    MONIT 460

    MONITOR 461

    NCPRECOV 462

    NOTIFY 465

    NTFYOP 466

    RECOVERY 469

RESIDENT control file entries in the control file 474

SESSION (SNA) statement for sessions monitored by

    AON/SNA 475

SUBSYSTEM control file entry for NVAS subsystem

    interface with SNA help desk 475

TCP390 control file entry 476

TGWSWITCH (SNA) 481

THRESHOLDS, setting 482

TIMER control file entries 486

TSOSERV 489

X25MONIT 490

AON Message Classes 466

AON/SNA feature 263

API statement 565

APPCCMD 386

Application Management Instrumentation

    enabling 263

application name

    CNM data transfer task 126

    NetView 149

ARCHIVE.ACCESSDELAY statement 8

ARCHIVE.BROWSE.DATASPACES statement 9  
 ARCHIVE.HLQ statement 10  
 ARCHIVE.INDEX.DATACLAS statement 11  
 ARCHIVE.INDEX.MGMTCLAS statement 12  
 ARCHIVE.INDEX.SPACE statement 14  
 ARCHIVE.INDEX.STORCLAS statement 15  
 ARCHIVE.INDEX.UNIT statement 11, 16  
 ARCHIVE.INDEX.VOLUME statement 17  
 ARCHIVE.MESSAGE.DATACLAS statement 19  
 ARCHIVE.MESSAGE.MGMTCLAS statement 20  
 ARCHIVE.MESSAGE.SPACE statement 21  
 ARCHIVE.MESSAGE.STORCLAS statement 22  
 ARCHIVE.MESSAGE.UNIT statement 18, 23  
 ARCHIVE.MESSAGE.VOLUMES statement 24  
 ARCHIVE.WRITE statement 25  
 ASSIGN command 296  
 ASSIGN statement 26  
 ASYNC\_TASKS statement 541  
 AT command 304  
 AT-TLS function 250  
 ATCCONxx  
     STATOPT statement 391  
 attack detection event 94  
 attack detection event, IDS 93  
 attack detection flood event 104  
 attack detection flood event, IDS 103  
 AUTH statement  
     usage 295  
 authorization checking  
     commands and command lists 301  
 authorize remote operators 386  
 authorized receiver 296, 324  
 AUTOCMD statement 27  
 AUTOFLIP keyword, LOGINIT statement 344  
 automatic re-activation 370  
 automating MSUs 175  
 automation operator names 416  
 automation table 27  
 automation task  
     operator 373  
 AUTOOPS 415  
 AUTORATE statement 175  
 AUTOTASK command 373  
 autotask statement 88  
 AUTOTASK statement 28  
 autotask, define to RODM access and control task 395  
 autotasks 28  
 auxInitCmd statement 29

## B

BACKUP keyword, DEFFOCPCT statement 315  
 banner statement 30  
 BER data 180  
 BINDFAIL keyword, KCLASS statement 339  
 BLOCK keyword  
     RMTSEC statement 387  
 block remote operators 386  
 BNJMBDST 1  
 BNJMBDST sample  
     DBFULL statement 313  
 books  
     see publications xv  
 BOUNDS keyword, PCLASS statement 377  
 BROWSE  
     CANZLOG group 85  
 BufEvtMaxSize statement 506

BufEvtNegRespLimit statement 506  
 BufEvtPath statement 507  
 BufEvtRdBlklen statement 508  
 BufEvtShrinkSize statement 509  
 buffer size definition  
     for reading data from event buffer 508  
 BufferEvents statement 510  
 BufferEventsLimit statement 511  
 BufferFlushRate statement 511  
 BUILD macroinstruction 391

## C

C (Command List) statement 300  
 CAF interfaces 58  
 CALRTCFG statement 512  
 Canzlog  
     archive 25  
     archive function 9, 10  
     access delay 8  
     data class 19  
     space allocation 21  
     storage class 22  
 CANZLOG  
     browse 85  
     group 85  
 CCDEF statement 30  
 CDLOG 419  
 CDRMDEF statement 148  
 CDRSC, VTAM definition statement 345  
 CELL\_POOLS statement 541  
 change bars xx  
 character translation sets 272, 273  
 CHARACTER\_VALIDATION statement 543  
 CHECKPOINT statement 566  
 CHECKPOINT\_FUNCTION statement 544  
 class  
     keep 337, 347  
     of service 312  
     performance 347  
 Classes, AON Messages 466  
 CLIENTLISTPORT statement 535  
 CLIENTPORT statement 535  
 CLOSE command 322  
 CMD keyword, RMTSEC statement 388  
 CMDCLASS statement 301  
 CMDDEF statement 301  
 CMDLINE statement 308  
 CMDMDL statement 309  
 CMDRCVR statement 309  
 CMDSYN keyword  
     CMDDEF statement 301  
 CMDSYN statement 310  
 CMSGCFG statement 513  
 CNM data transfer 125, 126  
 CNMCMD 1  
 CNMCMD sample  
     %INCLUDE statement 333  
     CMDDEF statement 301  
 CNMDVIPP data set 117  
 CNMDVIPS data set 117  
 CNMEDCST command list 225  
 CNMETACI 256  
 CNMFANGC module 325  
 CNMI interface 31  
 CNMI statement 31  
 CNMPOLCY 1

CNMSBAK1 232  
 CNMSCAT2 232  
 CNMSCNFT 1  
 CNMSCNFT sample  
     %INCLUDE statement 333  
     ACTION statement 292  
     CMDLINE statement 308  
     COLUMNHEAD statement 310  
     HELD statement 327  
     HOLDPCNT statement 329  
     HOLDWARN statement 330  
     IMDAREA statement 331  
     INDENT statement 335  
     LASTLINE statement 341  
     LOCKIND statement 343  
     MLINDENT statement 349  
     NOPREFIX statement 367  
     NORMAL statement 367  
     NORMQMAX statement 369  
     PREFIX statement 379  
     TITLE statement 396  
     TITLEDATE statement 398  
     TITLEDOMID statement 399  
     TITLEOPID statement 401  
     TITLESTAT statement 402  
     TITLETIME statement 404  
 CNMSJ002 sample 117  
 CNMSJ009 sample 117  
 CNMSJM01 150, 176, 256  
 CNMSPAN2 241  
 CNMSTIDS sample  
     IDS.Attack\_Cmd statement 93  
     IDS.Attack\_CmdType statement 94  
     IDS.Auto\_Interval statement 95  
     IDS.Auto\_Thresh statement 96  
     IDS.ClearStat\_Day statement 96  
     IDS.ClearStat\_Inform statement 97  
     IDS.ClearStat\_Log statement 98  
     IDS.ClearStat\_Log\_File statement 99  
     IDS.ClearStat\_Time statement 99  
     IDS.CONSOLEMSG statement 100  
     IDS.DSIPARM statement 100  
     IDS.Event\_Inform statement 101  
     IDS.Event\_Limit statement 102  
     IDS.Event\_Log statement 102  
     IDS.Event\_Log\_File statement 103  
     IDS.Flood\_Cmd statement 103  
     IDS.Flood\_CmdType statement 104  
     IDS.probeid statement 105  
     IDS.Report\_Cmd statement 106  
     IDS.Report\_CmdType statement 107  
     IDS.Report\_Inform statement 108  
     IDS.Report\_Log statement 108  
     IDS.Report\_Log\_File statement 109  
     IDS.Scan\_Cmd statement 109  
     IDS.Scan\_CmdType statement 110  
     IDS.SYSLOG.FILENAME statement 111  
     IDS.SYSLOGMSG statement 112  
     IDS.TCP\_Cmd statement 113  
     IDS.TCP\_CmdType statement 113  
     IDS.UDP\_Cmd statement 114  
     IDS.UDP\_CmdType statement 115  
 CNMSTPWD 2  
 CNMSTWBM member  
     webmenu statement 276  
 CNMSTYLE 2  
 CNMSTYLE sample 36

CNMSTYLE sample (*continued*)  
     .REXEC.PORT statement 209  
     .REXEC.SOCKETS statement 210  
     .REXEC.TCPNAME statement 211  
     %INCLUDE statement 333  
     ALCACHE statement 168  
     AlertRcvName statement 8  
     ARCHIVE.ACCESSDELAY statement 8  
     ARCHIVE.BROWSE.DATASPACES statement 9  
     ARCHIVE.HLQ statement 10  
     ARCHIVE.INDEX.BLOCKSIZE statement 11  
     ARCHIVE.INDEX.DATACLAS statement 11  
     ARCHIVE.INDEX.MGMTCLAS statement 12  
     ARCHIVE.INDEX.SPACE statement 14  
     ARCHIVE.INDEX.STORCLAS statement 15  
     ARCHIVE.INDEX.UNIT statement 16  
     ARCHIVE.INDEX.VOLUME statement 17  
     ARCHIVE.MESSAGE.BLOCKSIZE statement 18  
     ARCHIVE.MESSAGE.DATACLAS statement 19  
     ARCHIVE.MESSAGE.MGMTCLAS statement 20  
     ARCHIVE.MESSAGE.SPACE statement 21  
     ARCHIVE.MESSAGE.STORCLAS statement 22  
     ARCHIVE.MESSAGE.UNIT statement 23  
     ARCHIVE.MESSAGE.VOLUMES statement 24  
     ARCHIVE.WRITE statement 25  
     ASSIGN statement 26  
     AUTOCMD statement 27  
     AUTOTASK statement 28  
     auxInitCmd statement 29  
     banner statement 30  
     CCDEF statement 30  
     CNMI statement 31  
     COMMON statement 31  
     COMMON.CNMIP.DNSTimeout statement 32  
     COMMON.CNMSNMP.MIBPATH statement 33  
     COMMON.CNMSNMP.MIBS statement 34  
     COMMON.CNMSNMP.port statement 35  
     COMMON.CNMSNMP.retries statement 35  
     COMMON.CNMSNMP.timeout statement 35  
     COMMON.CNMTRAP.taskname.MAXTCPCONN  
         statement 37  
     COMMON.CNMTRAP.taskname.STACKNAME  
         statement 38  
     COMMON.CNMTRAP.taskname.TCPPORT statement 38  
     COMMON.CNMTRAP.taskname.TRACE statement 39  
     COMMON.CNMTRAP.taskname.UDPPORT statement 40  
     COMMON.DUIFHNAM statement 40  
     COMMON.DUIFHPRC statement 41  
     COMMON.EKGHNAM statement 41  
     COMMON.EKGHPRC statement 42  
     COMMON.EZLIMITDELAY statement 42  
     COMMON.EZLIPITraceJCLWait statement 42  
     COMMON.EZLRTMTIMER statement 43  
     COMMON.EZLsmtpHOSTNAME statement 43  
     COMMON.EZLsmtpNAME statement 44  
     COMMON.EZLTCPcTRACEwriter statement 44  
     COMMON.EZLTCPpTRACEwriter statement 45  
     COMMON.EZLTRACED statement 45  
     COMMON.FLC\_DEF\_NETW\_VIEW statement 46  
     COMMON.FLC\_DEFAULT\_ITNM\_OBJECT\_STATUS  
         statement 46  
     COMMON.FLC\_EXCEPTION\_VIEW\_FILE statement 47  
     COMMON.FLC\_RODMAPPL statement 47  
     COMMON.FLC\_RODMINT statement 48  
     COMMON.FLC\_RODMNAME statement 48  
     COMMON.FLC\_RODMRETRY statement 49  
     COMMON.FLC\_RUNCMDRETRY statement 49

CNMSTYLE sample (*continued*)

COMMON.FLC\_TCPNAME statement 49  
COMMON.IPPORTMON.INTVL statement 50  
COMMON.IPPORTMON.IPADD statement 50  
COMMON.IPPORTMON.PORTNUM statement 51  
COMMON.IPPORTMON.PORTTO statement 51  
COMMON.PKTS.INSTANCE.POOL statement 52  
COMMON.SMFVPD statement 52  
COMMON.STACKFAMILY statement 53  
COMMON.WAITTIME statement 53  
COMMON.XDOMTIME statement 54  
ConsFixed statement 54  
console mask 55  
CORRELATION statement 56  
DEFAULTS statement 58  
DISCOVERY.INTAPPL statement 59  
DISCOVERY.INTINTERFACE statement 60  
DISCOVERY.INTTELNET statement 60  
DISCOVERY.NetViewOnly statement 61  
DISCOVERY.SNMP statement 61  
DLA.cmdb\_ftp\_server statement 62  
DLA.debug 62  
DLA.ftp\_log\_blocksize 63  
DLA.ftp\_log\_filename 63  
DLA.ftp\_log\_space\_pri 64  
DLA.ftp\_log\_space\_sec 64  
DLA.ftp\_log\_unit 65  
DLA.ftp\_log\_volume 66  
DLA.ftp\_max\_xmit\_tm 66  
DLA.ftp\_remote\_dir 67  
DLA.ftp\_timeout 67  
DLA.ftp\_uid 68  
DLA.is\_second\_level 68  
DLA.statefile 69  
DLA.tsouser 69  
DLA.xml\_blocksize 70  
DLA.xml\_filename 70  
DLA.xml\_space\_pri 71  
DLA.xml\_space\_sec 72  
DLA.xml\_unit 72  
DLA.xml\_volume 73  
DOMAIN statement 73  
DVIPA.Event.Delay statement 74  
DVIPA.INTDVCONN statement 75  
DVIPA.INTDVDEF statement 76  
DVIPA.INTDVROUT statement 76  
DVIPA.INTDVTAD statement 75  
DVIPA.Mast.Disc.Delay statement 77  
DVIPA.Mast.EMARf.Delay statement 77  
DVIPA.STATS.DVIPA statement 78  
DVIPA.STATS.Logto statement 79  
DVIPA.STATS.PORT statement 80  
DVIPA.STATS.Pri.MAXR statement 80  
DVIPA.STATS.Sec.MAXR statement 81  
DVIPA.STATS.TCPNAME statement 81  
EMAAUTO statement 82  
endcmd.AutoTask statement 83  
endcmd.close.leeway statement 83  
ENT.CONNCHECK.INT statement 84  
ENT.GROUP.groupname statement 85  
ENT.INT.name statement 86  
ENT.SYSTEMS.name statement 87  
function.autotask statement 88  
GHB.TCPNAME statement 89  
HARDCOPY statement 90  
HLLENV statement 91  
idleParms statement 92

CNMSTYLE sample (*continued*)

IDS.Attack\_Cmd statement 93  
IDS.Attack\_CmdType statement 94  
IDS.Auto\_Intvl statement 95  
IDS.Auto\_Thresh statement 96  
IDS.ClearStat\_Day statement 96  
IDS.ClearStat\_Inform statement 97  
IDS.ClearStat\_Log statement 98  
IDS.ClearStat\_Log\_File statement 99  
IDS.ClearStat\_Time statement 99  
IDS.CONSOLEMSG statement 100  
IDS.DSIPARM statement 100  
IDS.Event\_Inform statement 101  
IDS.Event\_Limit statement 102  
IDS.Event\_Log statement 102  
IDS.Event\_Log\_File statement 103  
IDS.Flood\_Cmd statement 103  
IDS.Flood\_CmdType statement 104  
IDS.probeid statement 105  
IDS.Report\_Cmd statement 106  
IDS.Report\_CmdType statement 107  
IDS.Report\_Inform statement 108  
IDS.Report\_Log statement 108  
IDS.Report\_Log\_File statement 109  
IDS.Scan\_Cmd statement 109  
IDS.Scan\_CmdType statement 110  
IDS.SYSLOG.FILENAME statement 111  
IDS.SYSLOGMSG statement 112  
IDS.TCP\_Cmd statement 113  
IDS.TCP\_CmdType statement 113  
IDS.UDP\_Cmd statement 114  
IDS.UDP\_CmdType statement 115  
INFORM.POLICY.MEMBER statement 116  
INIT.DVIPASTATS statement 116  
IPv6Env statement 122  
MSG.TECROUTE statement 129  
MVSPARM.ActionDescCodes 130  
MVSPARM.Cmd.Designator 131  
MVSPARM.Cmd.Scope 132  
MVSPARM.DEFAUTH 132  
NACMD.DESTPPI statement 136  
NACMD.INTCONINACT statement 137  
NACMD.INTCONNACT statement 138  
NACMD.INTTHEALTH statement 138  
NACMD.INTSESSACT statement 139  
NACMD.LCLPPIRV statement 139  
NACMD.OPID statement 140  
NACMD.PERSIST statement 141  
NACMD.ROWSxxx statement 142  
NACMD.SUBNODE statement 143  
NACMD.WAITSECS statement 144  
NetID statement 145  
NLDM.AMLUTDLY statement 145  
NLDM.AUTHDOM statement 146  
NLDM.AUTHORIZ statement 147  
NLDM.CDRMDEF statement 148  
NLDM.CDTIME statement 149  
NLDM.DRDELAY statement 150  
NLDM.DSRBO statement 150  
NLDM.ERCOUNT statement 151  
NLDM.FCTIME statement 151  
NLDM.KEEPDISC statement 152  
NLDM.KEEPMEM statement 152  
NLDM.KEEPPIU statement 153  
NLDM.KEEPRTM statement 153  
NLDM.KEEPSESS statement 154  
NLDM.LOG statement 154

- CNMSTYLE sample (*continued*)
- NLDM.LUCOUNT statement 155
  - NLDM.MACRF statement 156
  - NLDM.MAXEND statement 157
  - NLDM.NETID statement 157
  - NLDM.PDDNM statement 158
  - NLDM.PERFMEM statement 158
  - NLDM.PIUTNUM statement 160
  - NLDM.PIUTSIZE statement 160
  - NLDM.RETRY statement 161
  - NLDM.RTDASD statement 161
  - NLDM.RTM statement 162
  - NLDM.RTMDISP statement 162
  - NLDM.SAW statement 163
  - NLDM.SAWNUM statement 163
  - NLDM.SAWSIZE statement 164
  - NLDM.SDDNM statement 164
  - NLDM.SESSMAX statement 165
  - NLDM.SESSTATS statement 165
  - NLDM.TRACEGW statement 166
  - NLDM.TRACELU statement 166
  - NLDM.TRACESC statement 167
  - NMCstatus.errorDSN statement 168
  - NPDA.ALERTFWD statement 170
  - NPDA.ALERTLOG statement 172
  - NPDA.ALRTINFP statement 173
  - NPDA.ALTER\_ALERT statement 174
  - NPDA.AUTORATE statement 175
  - NPDA.DSRBO statement 176
  - NPDA.DSRBU statement 177
  - NPDA.MACRF statement 178
  - NPDA.MDSIND statement 179
  - NPDA.NPDA.ERR\_RATE statement 177
  - NPDA.PDDNM statement 179
  - NPDA.PDFILTER statement 180
  - NPDA.PRELOAD\_BER statement 180
  - NPDA.R (Ratio) statement 181
  - NPDA.RATE statement 182
  - NPDA.REPORTS statement 183
  - NPDA.SDDNM statement 184
  - NPDA.TECROUTE statement 185
  - NPDA.W (Wrap) statement 185
  - NQNSUP.USE 188
  - NRM.CMODE statement 189
  - NRM.CONNINT statement 190
  - NRM.CONNRETRY statement 190
  - NRM.HBRETRY statement 191
  - NRM.HOSTDEST statement 191
  - NRM.IPV6SRC statement 192
  - NRM.PORT statement 193
  - NRM.RESET statement 194
  - NRM.SAMPLERATE statement 194
  - NRM.STATUS statement 195
  - NRM.TYPE statement 196
  - NRM.XCLDOM statement 196
  - NRM.XCLSRC statement 197
  - NRM.XCLTASKN statement 198
  - NRM.XCLTASKT statement 198
  - NVROLE statement 199
  - NVSP.srvrname.NUMTHRDS statement 200
  - NVSP.srvrname.PDS statement 200
  - NVSP.srvrname.PORT statement 201
  - NVSP.srvrname.SECURE statement 202
  - NVSP.srvrname.TRC statement 203
  - NVSP.srvrname.WAIT statement 204
  - OpDsPrefix statement 204
  - OPKT.STORAGE statement 205
- CNMSTYLE sample (*continued*)
- PEXLSTxx statement 159
  - PKTS.STORAGE statement 206
  - POLICY statement 206
  - PRINTPDU38 588
  - PWD statement 207
  - RCMRODMUser statement 208
  - RESTORE.CGLOBAL statement 208
  - REXX.CMDENV.name statement 211
  - REXX.FUNCPKGLIST.LCL.name statement 212
  - REXX.FUNCPKGLIST.SYS.name statement 213
  - REXX.FUNCPKGLIST.USR.name statement 213
  - RMTALIAS statement 214
  - RMTINIT.IP statement 216
  - RMTINIT.PORT statement 216
  - RMTINIT.SAFrefresh statement 217
  - RMTINIT.SECOPT statement 217
  - RMTINIT.SNA statement 218
  - RMTINIT.SOCKETS statement 218
  - RMTINIT.TCPNAME statement 219
  - RMTSYN statement 220
  - RODMname statement 222
  - ROTECDE statement 222
  - RRD statement 223
  - RSH.PORT statement 224
  - RSH.SOCKETS statement 224
  - RTNDEF.BASE.COLLECTOR.APPL 225
  - RTNDEF.BASE.COLLECTOR.STACK 225
  - RTNDEF.BASE.COLLECTOR.SYSTEM 226
  - RTNDEF.BASE.HB.APPL 226
  - RTNDEF.BASE.MEMBER 226
  - RTNDEF.BASE.PROCESSOR.APPL 227
  - RTNDEF.BASE.PROCESSOR.STACK 227
  - RTNDEF.BASE.PROCESSOR.SYSTEM 228
  - RTNDEF.BASE.PROVIDER 228
  - RTNDEF.COLLECTOR.APPL.NETVIEW 228
  - RTNDEF.PROCESSOR.STACK 229
  - RTT.PORT statement 229
  - RTT.SOCKETS statement 230
  - RTT.TCPNAME statement 230
  - SECOPTS.AUTHCHK statement 231
  - SECOPTS.CMDAUTH statement 231
  - SECOPTS.OPERSEC statement 234
  - SECOPTS.OPSPAN statement 238
  - SECOPTS.RMTAUTH statement 239
  - SECOPTS.SARESAUTH statement 240
  - SECOPTS.SPANAUTH statement 241
  - SECOPTS.SPANCHK statement 241
  - SECOPTS.SURROGAT statement 242
  - SECOPTS.WEBAUTH statement 242
  - SSI.PPI statement 243
  - SSI.ProcString statement 243
  - SSI.ReviseTable statement 244
  - styleMsg statement 245
  - styleMsg.wrap statement 245
  - stylevar statement 246
  - SuppChar statement 247
  - TAMEL.CONV.sysdef statement 248
  - TAMEL.PORT statement 249
  - TAMEL.SOCKETS statement 249
  - TAMEL.TCPNAME statement 250
  - TAMEL.TTLS statement 250
  - TAMEL.USETCPIP statement 251
  - TASK.taskname.INIT statement 252
  - TASK.taskname.MEM statement 253
  - TASK.taskname.MOD statement 253
  - TASK.taskname.PRI statement 254

CNMSTYLE sample (*continued*)  
 TCPCONN.DASD statement 255  
 TCPCONN.DSRBO statement 256  
 TCPCONN.ERRMSG.THROTTLE statement 256  
 TCPCONN.GTF.stackname statement 257  
 TCPCONN.HASHSIZE.stackname statement 257  
 TCPCONN.KEEP.stackname statement 258  
 TCPCONN.MACRF statement 259  
 TCPCONN.PDDNM statement 259  
 TCPCONN.ROWSA.stackname statement 260  
 TCPCONN.ROWSD.stackname statement 261  
 TCPCONN.SDDNM statement 261  
 TCPname statement 262  
 TCPserver statement 262  
 TOWER statement 263  
 TRACE.MODE statement 266  
 TRACE.MODFILT statement 267  
 TRACE.MONOPER statement 268  
 TRACE.OPTION statement 269  
 TRACE.SAFrc statement 270, 271  
 TRACE.SIZE statement 271  
 TRACE.TASK statement 272  
 transMember statement 272  
 transTbl statement 273  
 VTAMCP.USE statement 273  
 WBSORDER statement 274  
 WEB statement 275  
 webmenu statement 276  
 WEBSITE statement 278  
 WINDOW.EXEC statement 279  
 WLM.SubSystemName statement 279  
 XCF.GROUPNUM statement 279  
 XCF.MASTDVIPA statement 280  
 XCF.PROCSTRnn statement 281  
 XCF.RANK statement 282  
 XCF.TAKEOVER.CLIST statement 282  
 XCF.TAKEOVER.CONVIPrn statement 283  
 XCF.TAKEOVER.CONVSNAnn statement 284  
 XCF.TAKEOVERDELAY statement 284  
 XCF.TAKEOVER.DURATION statement 285  
 XCF.TAKEOVER.INITWAIT statement 286  
 XCF.TAKEOVER.NETCONVS statement 286  
 CNMSTYLE.NETID 145  
 CNMTAMEL 248  
 CNMTAMEL task 294  
 CNMTARG statement 125  
 color operands 332  
 COLTAG keyword, PREFIX statement 381  
 column heading line, command facility 310  
 COLUMNHEAD statement 310  
 command  
     authority verification 301  
     authorization 231  
     define 301  
     echoing to screen 304  
     module, loading at initialization 304  
     receiver, defining program-to-program interface 309  
     synonyms, defining 301  
     type 301  
     uppercase 306  
 command authority 231, 240  
 command facility  
     panel  
         action area 330  
         action messages 292  
         column heading line 310  
         command line 308

command facility (*continued*)  
 panel (*continued*)  
     domain name 399  
     held area 330  
     held messages 327  
     immediate messages 331  
     last line 341  
     lock indicator 343  
     normal messages 367, 369  
     operator identifier 401  
     percentage for message 329  
     status indicators 402  
     title line 396  
     title line date 398  
     title line time 404  
     panel customization 30  
 command line, command facility 308  
 command list  
     define 301  
     displaying 300  
     synonyms, creating 301  
 common global variables 31, 199  
 COMMON statement 31  
 common variable  
     CNMSTYLE.NETID 145  
 COMMON.CNMIP.DNSTimeout statement 32  
 COMMON.CNMSNMP.MIBPATH statement 33  
 COMMON.CNMSNMP.MIBS statement 34  
 COMMON.CNMSNMP.port statement 35  
 COMMON.CNMSNMP.retries statement 35  
 COMMON.CNMSNMP.timeout statement 35  
 COMMON.CNMSNMP trap statement 36  
 COMMON.CNMTRAP.taskname.CONFIGFILE statement 36  
 COMMON.CNMTRAP.taskname.MAXTCPConn  
     statement 37  
 COMMON.CNMTRAP.taskname.STACKNAME statement 38  
 COMMON.CNMTRAP.taskname.TCPPORT statement 38  
 COMMON.CNMTRAP.taskname.TRACE statement 39  
 COMMON.CNMTRAP.taskname.UDPPORT statement 40  
 COMMON.DUIFHNAM statement 40  
 COMMON.DUIFHPRC statement 41  
 COMMON.EKGHNAM statement 41  
 COMMON.EKGHPRC statement 42  
 COMMON.EZLIMITDELAY statement 42  
 COMMON.EZLIPTraceJCLWaitt statement 42  
 COMMON.EZLRMTIMER statement 43  
 COMMON.EZLsmtpHOSTNAME statement 43  
 COMMON.EZLsmtpNAME statement 44  
 COMMON.EZLTCPrTRACEwriter statement 44  
 COMMON.EZLTCPrTRACEwriter statement 45  
 COMMON.EZLTRACEd statement 45  
 COMMON.FLC\_DEF\_NETW\_VIEW statement 46  
 COMMON.FLC\_DEFAULT\_ITNM\_OBJECT\_STATUS  
     statement 46  
 COMMON.FLC\_EXCEPTION\_VIEW\_FILE statement 47  
 COMMON.FLC\_RODMAPPL statement 47  
 COMMON.FLC\_RODMINT statement 48  
 COMMON.FLC\_RODMNAME statement 48  
 COMMON.FLC\_RODMRETRY statement 49  
 COMMON.FLC\_RUNCMDRETRY statement 49  
 COMMON.FLC\_TCPNAME statement 49  
 COMMON.IPPORTMON.INTVL statement 50  
 COMMON.IPPORTMON.IPADD statement 50  
 COMMON.IPPORTMON.PORTNUM statement 51  
 COMMON.IPPORTMON.PORTTO statement 51  
 COMMON.PKTS.INSTANCE.POOL statement 52  
 COMMON.SMFVPD statement 52

COMMON.STACKFAMILY statement 53  
 COMMON.WAITTIME statement 53  
 COMMON.XDOMTIME statement 54  
 Community statement 514  
 COMNTESC statement 312  
 CONCURRENT\_USERS statement 545  
 CONNECT\_VIOLATION\_MESSAGE 546  
 ConnectionMode statement 514  
 ConsFixed statement 54  
 CONSNAME keyword, PROFILE statement 383  
 console mask 55  
 console name, setting 383  
 conventions  
     typeface xx  
 correlation service 535  
 CORRELATION statement 56  
 COS statement 312  
 CPCPSESS (SNA) 421  
 cross-domain session 149  
 cross-domain  
     authorization 146, 317  
     sessions  
         data retrieval 148  
         global control 223  
 CTCA0102 sample  
     AUTH statement 296

## D

DASD keyword, KCLASS statement 339  
 data buffers 160, 163, 164  
 data display, hardware monitor 174  
 DATA keyword, KCLASS statement 339  
 data services request blocks 177  
 database full, hardware monitor 313  
 DB2SEC statement 58  
 DBFULL statement 313  
 Dbserver task message queue 573  
 DDFGENERIC 422  
 DDFGROUP 424  
 DEFAULTS statement 58  
 DEFENTPT statement 313  
 DEFFOCPT statement 314  
 define  
     RODM to RODM access and control task 384  
 Defining AON Control File Entries 409, 436  
 defining the IP environment 122  
 definition  
     statements  
         NetView 289  
         RODM 541  
 definition statement format 4  
 detection event, attack 94  
 detection event, IDS attack 93  
 detection event, IDS scan 109  
 detection event, scan 110  
 DGROUP keyword, KCLASS statement 339  
 directory names, notation xxi  
 DISCOVERY  
     INTERFACES subtower 264  
         HIPERSOCKETS subtower 264, 266  
         OSA subtower 264, 266  
     TELNET subtower 264  
 DISCOVERY support 263  
 discovery, resource  
     RTNDEF.BASE.COLLECTOR.APPL 225  
     RTNDEF.BASE.COLLECTORSTACK 225

discovery, resource (*continued*)  
     RTNDEF.BASE.COLLECTOR.SYSTEM 226  
     RTNDEF.BASE.HB.APPL 226  
     RTNDEF.BASE.MEMBER 226  
     RTNDEF.BASE.PROCESSOR.APPL 227  
     RTNDEF.BASE.PROCESSORSTACK 227  
     RTNDEF.BASE.PROCESSORSYSTEM 228  
     RTNDEF.BASE.PROVIDER 228  
     RTNDEF.COLLECTOR.APPL.NETVIEW 228  
     RTNDEF.PROCESSORSTACK 229  
 discovery, resources 61  
 DISCOVERY.INTAPPL statement 59  
 DISCOVERY.INTINTERFACE statement 60  
 DISCOVERY.INTTELNET statement 60  
 DISCOVERY.NetViewOnly statement 61  
 DISCOVERY.SNMP statement 61  
 distributed DVIPA  
     local TCP/IP job name 81  
     statistical records 79  
     statistical records, number 80, 81  
 distributed DVIPA data discovery  
     starting 116  
 DLA.cmdb\_ftp\_server statement 62  
 DLA.debug 62  
 DLA.ftp\_log\_blocksize 63  
 DLA.ftp\_log\_filename 63  
 DLA.ftp\_log\_space\_pri 64  
 DLA.ftp\_log\_space\_sec 64  
 DLA.ftp\_log\_unit 65  
 DLA.ftp\_log\_volume 66  
 DLA.ftp\_max\_xmit\_tm 66  
 DLA.ftp\_remote\_dir 67  
 DLA.ftp\_timeout 67  
 DLA.ftp\_uid 68  
 DLA.is\_second\_level 68  
 DLA.statefile 69  
 DLA.tsouser 69  
 DLA.xml\_blocksize 70  
 DLA.xml\_filename 70  
 DLA.xml\_space\_pri 71  
 DLA.xml\_space\_sec 72  
 DLA.xml\_unit 72  
 DLA.xml\_volume 73  
 DOMAIN keyword  
     RMTSEC statement 388  
 domain keyword, RMTSYN statement 220  
 domain name  
     command facility panel 399  
 DOMAIN statement 73, 567  
 domain table 145  
 DOMAINS statement 317  
 DSCP 320  
 DS16INIT 4  
 DS16INIT sample  
     DEFENTPT statement 313  
     DEFFOCPT statement 314  
     RETRY statement 386  
 DSIALATD 2  
 DSIALATD sample  
     ALIASMEM statement 293  
 DSIALTAB 2  
 DSIALTAB sample  
     COS statement 312  
     LU statement 345  
     MODE statement 350  
     ORIGNET statement 374  
 DSIAMLTD 2

DSIAMLUT task 145  
 DSICMD sample 301, 309, 310, 312, 341, 375, 405  
 DSICNM 2  
     M (Maximum) statement 346  
 DSICNM sample  
     A (Alert) statement 289  
     C (Command List) statement 300  
     F (Filter) statement 324  
     O MONIT statement 370  
     O RESET statement 370  
     O SECSTAT statement 371  
     O SENDMSG statement 372  
     SENDMSG statement 388  
 DSICPINT 2  
 DSICRTR, CNM router subtask 321  
 DSICRTTD 2  
 DSICRTTD sample  
     DEFFOCPT statement 314  
 DSICTMOD sample 323  
 DSIDB2DF 2  
 DSIDB2DF sample  
     SUBSYSTEM statement 395  
 DSIELMEM 2  
 DSIELXIT 2  
 DSIHINIT 2  
 DSIHINIT sample  
     PARTNER statement 375  
     RETRY statement 386  
 DSIIILGCF 3  
 DSIKINIT 3  
 DSilogBK 3  
 DSilogBK sample  
     LOGINIT statement 344  
 DSILUCTD 3, 125  
 DSIOFF 3  
 DSIOFF sample  
     %INCLUDE statement 333  
     OPERATOR statement 373  
     PROFILEN statement 383  
 DSIPARM table 218  
 DSIPRAO sample  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPRFLC sample  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPRFMO sample  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPRFWD sample  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPRFxx 3  
 DSIPROFA 382  
 DSIPROFA sample  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPROFB 382  
 DSIPROFB sample 372  
     AUTH statement 295  
     PROFILE statement 382  
 DSIPROFx 3  
 DSIQTSKI 3  
 DSIQTSKI sample  
     CMDRCVR statement 309  
     REP statement 384  
     TASK statement 395  
 DSIREXCF 3  
 DSIREXCF sample  
     PROTOCOL statement 384  
 DSIRHOST 3  
 DSIROVSI 3  
 DSIRSHCF 3  
 DSIRSHCF sample  
     PROTOCOL statement 384  
 DSIRTTD 3  
 DSIRTTD sample  
     PORT statement 379  
     SOCKETS statement 389  
     TCPNAME statement 395  
 DSISECUR 3  
 DSISECUR sample, RMTSEC statement 386  
 DSISVRTD 3  
 DSITCPFC 3  
 DSITCPFC sample  
     PORT statement 379  
     SOCKETS statement 389  
     TCPNAME statement 395  
 DSITCPRF 3  
 DSITCPRF sample  
     NETCONV\_IP statement 352  
     WEB\_SERVER statement 407  
 DSITRCBK 3  
 DSITRCBK sample  
     LOGINIT statement 344  
 DSIVPARM 4  
 DSIVPARM sample  
     VPDINIT statement 406  
 DSIWBMEM 4  
 DSIWBMEM sample  
     PORT statement 379  
     SOCKETS statement 389  
     TCPNAME statement 395  
 DSIWLS macro 322  
 DSPLYLOC keyword, PCLASS statement 378  
 DSRB 321  
 DSRBO 150, 176, 256  
 DSRBO keyword, DSTINIT statement 321  
 DSRBU 177  
 DSRBU keyword, DSTINIT statement 321  
 DST initialization  
     exit routine  
         DSIAINIT, defined as 293  
         DSILINIT, defined as 125, 148  
         keywords 318  
 DSTINIT statement 318  
 DUIDGHB task  
     TCPNAME statement 89  
 DUILFCASB calls 568  
 DUILPMEM 4  
 DUILPMEM keyword, AMELINIT statement 294  
 DUILPMEM sample  
     TCPNAME statement 395  
 DUIGINIT 4  
 DUIGINIT sample  
     API statement 565  
     CHECKPOINT statement 566  
     DOMAIN statement 567  
     GMTOFFSET statement 567  
     JAPANESE statement 568  
     LCON-AGGRST-REQUIRED statement 569  
     LCON-ASSOCIATE-NULL-NODE-WITH-LINK  
         statement 571  
     LCON-HEX-SUBVECTOR-DISPLAY statement 572

DUIGINIT sample (*continued*)  
 LCON-MAX-LOCATE-RESOURCE-VIEWS statement 572  
 LCON-MAX-QUEUE-DBSERVER statement 573  
 LCON-MAX-QUEUE-EVENTMGR statement 574  
 LCON-MAX-QUEUE-IPC statement 574  
 LCON-MAX-QUEUE-IRMGR statement 575  
 LCON-MAX-QUEUE-MAINTASK statement 576  
 LCON-MAX-QUEUE-NETCMD statement 577  
 LCON-MAX-QUEUE-NETCON statement 577  
 LCON-MAX-QUEUE-OPERIF statement 578  
 LCON-MAX-QUEUE-RCMGR statement 579  
 LCON-MAX-QUEUE-VIEWMGR statement 580  
 LCON-MAX-QUEUE-VSTATMGR statement 581  
 LCON-NCC-RETRY-LIMIT statement 582  
 LCON-NCC-RSC-LIMIT statement 583  
 LCON-NMG-POLL-INTERVAL statement 583  
 LCON-OPERATOR-CMD-AUDIT statement 584  
 LCON-REPORT-UNKNOWN-STATUS statement 585  
 LCON-STATUS-DELAY-MAX statement 586  
 LCON-STATUS-DELAY-TIME statement 587  
 LEVEL statement 587  
 RODMID statement 590  
 RODMNAME statement 590  
 STORAGE statement 591  
 TASK statement 591  
 TRACE statement 592  
 TRACEBYTES statement 593  
 TRACEPAGES statement 594  
 TYPE statement 594  
 DUIIGHB 4  
 DUIIGHB sample  
 TCPANAME statement 395  
 DUISFP 4  
 DUISFP sample  
 AMELINIT statement 294  
 DUMP\_FOR\_BAD\_USER\_DATA statement 546  
 DUMP\_LIMIT statement 547  
 DUMP\_SCOPE statement 548  
 DVIPA  
     connection routing data 76  
     connections data 75  
     data 76  
     distributed 78  
     distributor targets 75  
     DVCONN subtower 264  
     DVROUT subtower 264  
     DVTAD subtower 264  
     event, delay 74, 77  
     port 80  
     sysplex distributors data 75  
 DVIPA connections 261  
 DVIPA support 264  
 DVIPA.Event.Delay statement 74  
 DVIPA.INTDVCONN statement 75  
 DVIPA.INTDVDEF statement 76  
 DVIPA.INTDVROUT statement 76  
 DVIPA.INTDVTAD statement 75  
 DVIPA.Mast.Disc.Delay statement 77  
 DVIPA.Mast.EMARf.Delay statement 77  
 DVIPA.STATS.DVIPA statement 78  
 DVIPA.STATS.Logto statement 79  
 DVIPA.STATS.PORT statement 80  
 DVIPA.STATS.Pri.MAXR statement 80  
 DVIPA.STATS.Sec.MAXR statement 81  
 DVIPA.STATS.TCPNAME statement 81

## E

E/AS 129, 185  
 EBCDIC character set 272, 273  
 ECHO keyword  
 CMDDEF statement 301  
 education  
     see Tivoli technical training xix  
 EKGUST 4  
 EKGUST sample  
     ASYNC\_TASKS statement 541  
     CELL\_POOLS statement 541  
     CHARACTER\_VALIDATION statement 543  
     CHECKPOINT\_FUNCTION statement 544  
     CONCURRENT\_USERS statement 545  
     CONNECT\_VIOLATION\_MESSAGE statement 546  
     DUMP\_FOR\_BAD\_USER\_DATA statement 546  
     DUMP\_LIMIT statement 547  
     DUMP\_SCOPE statement 548  
     EXTEND\_HEAP\_SIZE statement 549  
     IO\_QUEUE\_THRESHOLD statement 549  
     LOG\_LEVEL statement 550  
     MAX\_CHUNK statement 550  
     MAX\_SEGMENT\_NUM 551  
     MAX\_WINDOW\_NUM statement 551  
     MLOG\_LEVEL statement 552  
     MTRACE\_TYPE statement 553  
     PLI\_ISA statement 554  
     PRIMARY\_HEAP\_SIZE statement 554  
     QUIES\_WAIT\_TASK\_TIME statement 555  
     QUIES\_WAIT\_USER\_TIME statement 556  
     SEC\_CLASS statement 556  
     SEC\_RNAME statement 558  
     SEGMENT\_POCKETS statement 559  
     SSB\_CHAIN statement 560  
     TRANSPARENT\_CHECKPOINT statement 560  
     WAIT\_ALLOC\_TIME statement 561  
     WINDOW\_CHKPT\_TIME statement 562  
     WINDOW\_POCKETS statement 562  
 EMAAUTO statement 82  
 EMCs  
     console name 55  
 EMCS consoles 132  
 END statement 5, 324  
 endcmd.AutoTask statement 83  
 endcmd.close.leeway statement 83  
 ending processing 324  
 ENDTASK command 386  
 ENDTASK processing  
     security 217  
 ENT.CONNCHECK.INT statement 84  
 ENT.GROUP.groupname statement 85  
 ENT.INT.name statement 86  
 ENT.SYSTEMS.name statement 87  
 enterprise  
     master switch 86  
     pacing interval for data 86  
     sysplex entry point 87  
 Enterpriseoid statement 515  
 ENVIRON AIP 424  
 ENVIRON DDF 425  
 ENVIRON EXIT 427  
 ENVIRON RACF 429  
 ENVIRON SETUP 431  
 ENVIRON TIMEOUT 434  
 environment variables, notation xxi  
 EPONLY keyword, DEFENTPT statement 314  
 ER data 161

ER keyword, MAPSESS statement 349  
 ERCVCFG statement 516  
 ERRMSG 256  
 error-to-traffic ratio value 181  
 EV keyword, W (Wrap) statement 186  
 event  
     attack detection 94  
     attack detection flood detection 104  
     IDS attack detection 93  
     IDS attack detection flood 103  
     IDS scan detection 109  
     scan detection 110  
     TCP traffic regulation 113  
     UDP traffic regulation 114, 115  
 event log file 103  
 event logging 173, 182  
 Event/Automation Service 129, 185, 501  
 Event/Automation Service definition statements 501  
 EventMaxSize statement 517  
 EVENTMGR task message queue 574  
 events  
     inform policy 101  
 EVERY command 304  
 exception list  
     NLDM PURGE command 159  
 ExceptionViewFilter 571  
 ExceptionViewList 571  
 exit routine 148  
 EXTEND\_HEAP\_SIZE statement 549  
 extended console name 55  
 EZLCFG01 4  
 EZLEMAIL 44  
 EZLESMT 44  
 EZLINSMP 4  
     IDS inform policy definition 98, 101  
 EZLTLOG 435

## F

F (Filter) statement 324  
 FailbackValue statement 517  
 FAILURES keyword, KCLASS statement 339  
 file  
     event log 103  
     IDS probe statistics 109  
     probe summary statistics 99  
     UNIX System Services 111  
 Filter statement 518  
 FilterCache statement 519  
 FilterMode statement 520  
 filters  
     initial 180  
 FKRREP 109  
 flow control data request 151  
 focal point nodes, defining 314  
 FOCALPT command, .ALERTFWD settings 171  
 FOLDUP keyword  
     CMDDEF statement 301  
 FORCE command 341  
 FORWARD FOCALPT 437  
 full hardware monitor database 313  
 FULLSESS 437  
 FUNCT keyword, DSTINIT statement 320  
 function.autotask statement 88

## G

GATEWAY 438  
 GDPS Active/Active Continuous Availability solution  
     ENT.CONNCHECK.INT statement 84  
 GETCONID  
     command 133  
     console name 383  
 GHB.TCPNAME statement 89  
 global variable  
     common 31  
     DOMAIN 73  
     NetID 145  
     OpDsPrefix 204  
     RCMRODMUser 208  
     RODMname 222  
     TCPname 262  
 global variables 199  
 GMFALERT keyword, W (Wrap) statement 188  
 GMFHIS initialization statements  
     API statement 565  
     CHECKPOINT statement 566  
     DOMAIN statement 567  
     GMTOFFSET statement 567  
     JAPANESE statement 568  
     LCON-AGG-BUNDLE-INTERVAL statement 568  
     LCON-AGGRST-REQUIRED statement 569  
     LCON-AIP-RESET-INTERVAL statement 569  
     LCON-ALERT-CMD-TIMEOUT statement 570  
     LCON-ASSOCIATE-NODE-WITH-LINK  
         statement 571  
     LCON-EVCHANGE-BUFFER-INTERVAL statement 571  
     LCON-HEX-SUBVECTOR-DISPLAY statement 572  
     LCON-MAX-LOCATE-RESOURCE-VIEWS statement 572  
     LCON-MAX-QUEUE-DBSERVER statement 573  
     LCON-MAX-QUEUE-EVENTMGR statement 574  
     LCON-MAX-QUEUE-IPC statement 574  
     LCON-MAX-QUEUE-IRMGR statement 575  
     LCON-MAX-QUEUE-MAINTASK statement 576  
     LCON-MAX-QUEUE-NETCMD statement 577  
     LCON-MAX-QUEUE-NETCON statement 577  
     LCON-MAX-QUEUE-OPERIF statement 578  
     LCON-MAX-QUEUE-RCMGR statement 579  
     LCON-MAX-QUEUE-RTMGR statement 580  
     LCON-MAX-QUEUE-VIEWMGR statement 580  
     LCON-MAX-QUEUE-VSTATMGR statement 581  
     LCON-NCC-RETRY-LIMIT statement 582  
     LCON-NCC-RSC-LIMIT statement 583  
     LCON-NMG-POLL-INTERVAL statement 583  
     LCON-OPERATOR-CMD-AUDIT statement 584  
     LCON-REPORT-UNKNOWN-STATUS statement 585  
     LCON-SNATM-TIMEOUT statement 585  
     LCON-STATUS-DELAY-MAX statement 586  
     LCON-STATUS-DELAY-TIME statement 587  
     LEVEL statement 587  
     PRINTPDU38 588  
     RODMID statement 590  
     RODMNAME statement 590  
     STORAGE statement 591  
     TASK statement 591  
     TRACE statement 592  
     TRACEBYTES statement 593  
     TRACEPAGES statement 594  
     TYPE statement 594  
 GMFHIS time out 585  
 GMTOFFSET statement 567  
 graphics  
     enabling 264

## H

hardcopy printers, defining 90  
HARDCOPY statement 90  
hardware monitor  
  BER data 180  
  data display 174  
  database  
    event logging 173, 182  
    filters 180  
    statistic logging 173  
    wrap counts, specifying 185  
  enabling 265  
  focal point nodes 314  
  NPDA.LQTHRESH statement 180  
    report record logging 183  
HCL keyword, PROFILE statement 382  
held area, command facility panel 330  
held messages, command facility panel 327  
HELD statement 327  
high performance transport conversations 375  
highlight operands 332  
HLLENV statement 91  
HOLDPCNT statement 329  
HOLDWARN statement 330  
Hostname statement 521

## I

IC keyword  
  PROFILE statement 382  
identifier, operator 373  
idleParms statement 92  
IDS  
  attack detection event 93, 94  
  attack detection flood event 103, 104  
  DSIPARM data set 100  
  event log file 103  
  event logging 102  
  event source 100, 112  
  IDS scan detection event 109  
  inform policy 97, 101, 108  
  probe statistics log file 99  
  probe statistics report 106, 107  
  probe summary statistics, clearing 96, 98, 99  
  probeid 105  
  report log file 109  
  report logging 108  
  scan detection event 110  
  TCP traffic regulation event 113  
  threshold, defining 95  
  threshold, IDS events 96  
  UDP regulation event 115  
  UDP traffic regulation event 114  
  UNIX System Services file 111  
IDS attack detection event 93  
IDS attack detection flood event 103  
IDS report log file 109  
IDS scan detection event 109  
IDS.Attack\_Cmd statement 93  
IDS.Attack\_CmdType statement 94  
IDS.Auto\_Intvl statement 95  
IDS.Auto\_Thresh statement 96  
IDS.ClearStat\_Day statement 96  
IDS.ClearStat\_Inform statement 97  
IDS.ClearStat\_Log statement 98  
IDS.ClearStat\_Log\_File statement 99

IDS.ClearStat\_Time statement 99  
IDS.CONSOLEMSG statement 100  
IDS.DSIPARM statement 100  
IDS.Event\_Inform statement 101  
IDS.Event\_Limit statement 102  
IDS.Event\_Log statement 102  
IDS.Event\_Log\_File statement 103  
IDS.Flood\_Cmd statement 103  
IDS.Flood\_CmdType statement 104  
IDS.probeid statement 105  
IDS.Report\_Cmd statement 106  
IDS.Report\_CmdType statement 107  
IDS.Report\_Inform statement 108  
IDS.Report\_Log statement 108  
IDS.Report\_Log\_File statement 109  
IDS.Scan\_Cmd statement 109  
IDS.Scan\_CmdType statement 110  
IDS.SYSLOG.FILENAME statement 111  
IDS.SYSLOGMSG statement 112  
IDS.TCP\_Cmd statement 113  
IDS.TCP\_CmdType statement 113  
IDS.UDP\_Cmd statement 114  
IDS.UDP\_CmdType statement 115  
IGNRLSUP keyword  
  CMDDEF statement 301  
IHSAEVNT job 501  
IMDAREA statement 331  
immediate messages, command facility panel 331  
INCLUDE (%INCLUDE) statement 333  
INDENT statement 335  
indentation characteristics for messages  
  multiline 349  
  single text 335  
inform policy  
  IDS event 101  
  IDS probe statistics 108  
  loading at initialization 116  
  probe summary statistics, clearing 97  
inform policy member 493  
  CONTACT 500  
  GROUP 499  
  INFORM 499  
  SETUP 497  
INFORM.POLICY.MEMBER statement 116  
INIT.DVIPASTATS statement 116  
INIT.EMAAUTO statement 117  
INIT.NRM statement 117  
INIT.OPKT statement 118  
INIT.PKTS statement 119  
INIT.TCPCONN statement 119  
INIT.TIMER statement 120  
INITFAIL keyword, KCLASS statement 339  
initial  
  ratio values 181  
  span of control 336  
initial status of resources 370  
initialization  
  CNMTAMEL task 294  
  command module, loading at 304  
keywords  
  DST 318  
  NetView sequential log 344  
  network log 344  
  trace log 344  
  VPDTASK 406  
  record logging at 183  
installation exits 124

INSTALLOPT 440  
 inStore statement 120  
 intensity operands 333  
 interconnected network 312, 345, 350  
 intrusion detection services 100  
 Intrusion Detection Services 263  
 IO\_QUEUE\_THRESHOLD statement 549  
 IP environment, defining 122  
 IP management 264  
 IP packet trace management  
     starting 119  
 IP\_host keyword, RMTSYN statement 220  
 IPC task message queue 574  
 IPCONN 442  
 IPHOST 444  
 IPINFC 446  
 IPLOG statement 121  
 IPMGT  
     ACTMON subtower 264  
     IDS subtower 264  
 IPNAMESERV 448  
 IPPORT 450  
 IROUTER 453  
 IPTELNET 455  
 IPTN3270 456  
 IPV4 keyword, RMTSYN statement 220  
 IPV6 keyword, RMTSYN statement 220  
 IPv6Env statement 122  
 IRMGR task message queue 575  
 ISPLAN statement  
     definition 336  
 verbose statement 122

## J

Japanese character translation 272, 273  
 JAPANESE statement 568  
 JesJobLog statement 123  
 JLOG statement 536

## K

kanji, character set 272, 273  
 KCLASS keyword, MAPSESS statement 348  
 KCLASS statement 337  
 keep class  
     defining 337  
     mapping sessions into 337, 347  
 keep class definitions 152  
 KEEPALIVE keyword, RMTINIT statement 219  
 KEEPIIU keyword  
     KCLASS statement 340  
 KEEPSESS keyword  
     KCLASS statement 340  
 KEYCLASS statement 341  
 keywords  
     \*PRE, KCLASS statement 340  
     \*SEC, KCLASS statement 340  
     ACBNAME, VPDINIT statement 406  
     AL, W (Wrap) statement 186  
     alias, RMTALIAS statement 214  
     AUTOFLIP, LOGINIT statement 344  
     BACKUP, DEFOCPT statement 315  
     BINDFAIL, KCLASS statement 339  
     BLOCK  
         RMTSEC statement 387

keywords (*continued*)  
     BOUNDS, PCLASS statement 377  
     CMD, RMTSEC statement 388  
     CMDSYN  
         CMDDEF statement 301  
     CNMCON  
         AUTH statement 295  
     CONSNAM, PROFILE statement 383  
     DASD, KCLASS statement 339  
     DATA, KCLASS statement 339  
     defining synonyms for command 301  
     DGROUP, KCLASS statement 339  
     DOMAIN  
         RMTSEC statement 388  
     domain, RMTSYN statement 220  
     DSPLYLOC, PCLASS statement 378  
     DSRBO, DSTINIT statement 321  
     DSRBU, DSTINIT statement 321  
     DUIFPMEM, AMELINIT statement 294  
     ECHO  
         CMDDEF statement 301  
     EPONLY, DEFENTPT statement 314  
     ER, MAPSESS statement 349  
     EV, NPDA.W (Wrap) statement 185  
     FAILURES, KCLASS statement 339  
     FUNCT, DSTINIT statement 320  
     GMFAALERT, W (Wrap) statement 188  
     HCL, PROFILE statement 382  
     IC  
         PROFILE statement 382  
     IDS.Attack\_Cmd statement 93  
     IDS.Attack\_CmdType statement 94  
     IDS.ClearStat\_Inform statement 97  
     IDS.Event\_Inform statement 101  
     IDS.Flood\_Cmd statement 103  
     IDS.probeid statement 105  
     IDS.Report\_Cmd statement 106  
     IDS.Report\_Inform statement 108  
     IDS.Scan\_Cmd statement 109  
     IDS.Scan\_CmdType statement 110  
     IDS.TCP\_Cmd statement 113  
     IDS.TCP\_CmdType statement 113  
     IDS.UDP\_Cmd statement 114  
     IDS.UDP\_CmdType statement 115  
     IGNRLSUP  
         CMDDEF statement 301  
     INITFAIL, KCLASS statement 339  
     IP\_host, RMTSYN statement 220  
     IPV4, RMTSYN statement 220  
     IPV6, RMTSYN statement 220  
     IPv6Env statement 122  
     KCLASS, MAPSESS statement 348  
     KEEPALIVE, RMTINIT statement 219  
     KEEPIIU, KCLASS statement 340  
     KEEPSESS, KCLASS statement 340  
     LU  
         CNMTARG statement 125  
         RMTSEC statement 387  
     LUC.CTL 126  
     LUC.MAXSESS 126  
     LUC.PERSIST 126  
     MACREF, DSTINIT statement 322  
     MOD  
         CMDDEF statement 301  
     MSGRECVR, AUTH statement 295  
     MSU\_CORR, W (Wrap) statement 187

*keywords (continued)*

NAME  
  PARTNER statement 376  
NET, RMTSEC statement 387  
NETID, PARTNER statement 376  
netid, RMTSYN statement 220  
NGMFADMIN, AUTH statement 295  
NGMFVSPN, AUTH statement 295  
NLDM.AMLUTDLY statement 145  
NLDM.AUTHDOM statement 146  
NLDM.AUTHORIZ statement 147  
NOACTY, STATOPT statement 394  
NOMONIT, STATOPT statement 394  
OBJPCT, PCLASS statement 378  
OBJTIME, PCLASS statement 378  
OFFNPDA. REPORTS statement 184  
OMIT, STATOPT statement 393  
ON, NPDA.REPORTS statement 184  
PARMSYN  
  CMDDEF statement 301  
PARSE  
  CMDDEF statement 301  
PASS, RMTSEC statement 387  
PASSWORD  
  OPERATOR statement 373  
  VPDINIT statement 406  
PBSDN, DSTINIT statement 320  
PCLASS, MAPSESS statement 348  
PDDNM, DSTINIT statement 320  
PERSIST  
  DSTINIT statement 323  
  PARTNER statement 376  
PEXLSTxx statement 159  
PPASS, DSTINIT statement 321  
PRI  
  KCLASS statement 340  
  MAPSESS statement 348  
PRIMARY, DEFFOCP statement 316  
RENAME, KCLASS statement 339  
RES\_CORR, W (Wrap) statement 187  
RESIDENT  
  CMDDEF statement 301  
RESUME, LOGINIT statement 344  
RETRY  
  DSTINIT statement 323  
  NLDM.CDRMDEF statement 148  
RMTOP, RMTSEC statement 387  
RTDEF, PCLASS statement 377  
RTMDATA, KCLASS statement 339  
SARefresh, RMTINIT statement 217  
SARefresh,SNA statement 219  
SAW, KCLASS statement 340  
SBSDN, DSTINIT statement 321  
SDDNM, DSTINIT statement 321  
SEC  
  KCLASS statement 340  
  MAPSESS statement 348, 349  
SECURITY  
  CMDDEF statement 301  
SESSFAIL, KCLASS statement 339  
SESNORM, KCLASS statement 339  
SNA, RMTSYN statement 221  
SNAPRQ, VPDINIT statement 406  
SPASS, DSTINIT statement 321  
ST, W (Wrap) statement 187  
TARGOP, RMTSEC statement 387  
TP, MAPSESS statement 349

*keywords (continued)*

TRACDATA, KCLASS statement 339  
TYPE  
  CMDDEF statement 301  
  DEFFOCP statement 314, 315  
UNSQL, DSTINIT statement 320  
VPDREQ, VPDINIT statement 406  
VPDSTOR, VPDINIT statement 407  
VPDWAIT, VPDINIT statement 406  
VR, MAPSESS statement 349  
WRAPCNT, NPDA.ALCACHE statement 168  
XITBN, DSTINIT statement 321  
XITBO, DSTINIT statement 322  
XITCI, DSTINIT statement 322  
XITCO, DSTINIT statement 322  
XITDI, DSTINIT statement 322  
XITVI, DSTINIT statement 322  
XITVN, DSTINIT statement 322  
XITVO, DSTINIT statement 322  
XITXL, DSTINIT statement 322  
XLO, REPORTS statement 184

**L**

labels 4  
last line, command facility panel 341  
LASTLINE statement 341  
LBUILD macroinstruction 391  
LCLPORT statement 537  
LCON-AGG-BUNDLE-INTERVAL statement 568  
LCON-AGGRST-REQUIRED statement 569  
LCON-AIP-RESET-INTERVAL statement 569  
LCON-ALERT-CMD-TIMEOUT statement 570  
LCON-ASSOCIATE-NULL-NODE-WITH-LINK statement 571  
LCON-EVCHANGE-BUFFER-INTERVAL statement 571  
LCON-HEX-SUBVECTOR-DISPLAY statement 572  
LCON-MAX-LOCATE-RESOURCE-VIEWS statement 572  
LCON-MAX-QUEUE-DBSERVER statement 573  
LCON-MAX-QUEUE-EVENTMGR statement 574  
LCON-MAX-QUEUE-IPC statement 574  
LCON-MAX-QUEUE-IRMGR statement 575  
LCON-MAX-QUEUE-MAINTASK statement 576  
LCON-MAX-QUEUE-NETCMD statement 577  
LCON-MAX-QUEUE-NETCON statement 577  
LCON-MAX-QUEUE-OPERIF statement 578  
LCON-MAX-QUEUE-RCMGR statement 579  
LCON-MAX-QUEUE-RTMGR statement 580  
LCON-MAX-QUEUE-VIEWMGR statement 580  
LCON-MAX-QUEUE-VSTATMGR statement 581  
LCON-NCC-RETRY-LIMIT statement 582  
LCON-NCC-RSC-LIMIT statement 583  
LCON-NMG-POLL-INTERVAL statement 583  
LCON-OPERATOR-CMD-AUDIT statement 584  
LCON-REPORT-UNKNOWN-STATUS statement 585  
LCON-SNATM-TIMEOUT statement 585  
LCON-STATUS-DELAY-MAX statement 586  
LCON-STATUS-DELAY-TIME statement 587  
LEVEL statement 587  
lines of unequal speeds 481  
LOADEXIT statement 123  
loading installation exits 123  
local shared resources 322  
locate resource request 585  
lock indicator, command facility panel 343  
LOCKIND statement 343  
log file, event 103  
LOG\_LEVEL statement 550

logging vital product data 406  
logical unit (LU) name  
  CNMTARG statement 125  
names  
  defining 125  
  translating 345  
session authorization 125, 126

logical units 155  
LOGINIT statement 344

logon  
  mode names 350  
logon authority 234

logon panel 30  
LOGONPW statement 125

LSTHRESH 459

LU communications 375  
  full network qualified names 188  
  tracing 166

LU keyword  
  CNMTARG statement 126  
  RMTSEC statement 387

LU statement 345

LUC statement 125  
LUC.CNMTARG statement 125  
LUC.CTL statement 126

LUC.MAXSESS statement 126

LUC.PERSIST statement 126

LUDRSTAT command 442

## M

M (Maximum) statement 346

MACRF 156, 178, 259

MACRF keyword, DSTINIT statement 322

MAINTASK task message queue 576  
manuals

  see publications xv

MAPSESS statement 347

MAX\_CHUNK statement 550

MAX\_SEGMENT\_NUM 551

MAX\_WINDOW\_NUM statement 551

memStore statement 127

message

  authorized receiver 296

  BNJ045I 183

  buffer, parsing 304

  DSI151I 329, 330

  indicator

    classes 289, 324

    defining alarm attribute 290

    defining color attribute 289

    defining highlighting attribute 290

  prefix information 379

  unsolicited 295

Message Classes 466

messages

  translation 272

messages (MVS), changing attributes 351

method

  object independent 569

  resource traits change 571

MLINDENT statement 349

MLOG\_LEVEL statement 552

MOD keyword

  CMDDEF statement 301

mode names, logon 350

MODE statement 350

MODIFY.TOWER statement 128

MONIT 460

MONIT STOP command 370

MONITOR control file entry 461

MONOFF command 370

MS transport 273

MSG.TECROUTE statement 129

MSGCFG statement 521

MSGRECV keyword, AUTH statement 296

MSU\_CORR keyword, W (Wrap) statement 187

MSUs, automating 175

MTRACE\_TYPE statement 553

MultiSystem Manager

  enabling 264

MVS

  action descriptor codes 130

  command designator 131

  extended console support 132

MVS command

  console name 383

MVS command management

  enabling 265

MVS extended console name 55

MVS messages

  automation 134

  broadcast 136

MVS messages, changing attributes 351

MVS statement 351

MVSPARM statement 132

MVSPARM.ActionDescCodes statement 130

MVSPARM.Cmd.Designator statement 131

MVSPARM.Cmd.Scope statement 132

MVSPARM.Msg.Automation statement 134

MVSPARM.Msg.Automation.MaxAge statement 134

MVSPARM.Msg.Automation.Oldtag statement 135

MVSPARM.OperRecvBrdcst statement 136

## N

NACMD. INTSESSACT statement 139

NACMD.DESTPPI statement 136

NACMD.INTCONINACT statement 137

NACMD.INTCONNACT statement 138

NACMD.INTHEALTH statement 138

NACMD.LCLPPIRV statement 139

NACMD.OPID statement 140

NACMD.PERSIST statement 141

NACMD.ROWSxxx statement 142

NACMD.SUBNODE statement 143

NACMD.WAITSECS statement 144

NAME keyword

  PARTNER statement 376

NCP activation

  gateway tracing 166

NCPRECOV (SNA) 462

NET keyword, RMTSEC statement 387

NETCMD task message queue 577

NETCMDS class 232

NETCON task message queue 577

NETCONV TCP/IP sessions 250

NETCONV\_IP statement 352

NETID keyword

  PARTNER statement 376

netid keyword, RMTSYN statement 220

NetID statement 145

NetSP Single Logon Coordinator 236

NetView  
   application name 148  
   banner 30  
   defaults 58  
   resource discovery 61  
   samples, location 1  
   sequential log 344  
 NetView 3270 management console  
   definition statement  
     PORT 379  
     PROTOCOL 384  
     SOCKETS 389  
     TCPNAME 395  
     WINDOW 408  
 NetView application definition and status data 59  
 NetView for z/OS Enterprise Management Agent  
   EMAAUTO statement 82  
   NACMD.DESTPPI statement 136  
   NACMD.INTCONINACT statement 137  
   NACMD.INTCONNACT statement 138  
   NACMD.INTHEALTH statement 138  
   NACMD.INTSESSACT statement 139  
   NACMD.LCLPPIRV statement 139  
   NACMD.OPID statement 140  
   NACMD.PERSIST statement 141  
   NACMD.ROWSxxx statement 142  
   NACMD.SUBNODE statement 143  
   NACMD.WAITSECS statement 144  
   starting 117  
 NetView health data 138  
 NetView Resource Manager  
   starting 117  
 NetView web server  
   definition statement  
     PORT 379  
     SOCKETS 389  
     TCPNAME 395  
     WINDOW 408  
 NetViewAlertReceiver statement 522  
 network  
   interconnected 312  
   log, initial keywords 344  
   name, defining 374  
   status updates, receiving 371  
 Network Security Program 236  
 NEWLINE keyword, INDENT statement 335  
 NGMFADMN keyword, AUTH statement 297  
 NGMFCMDS keyword, AUTH statement 297  
 NGMFVSPN keyword, AUTH statement 297  
 NLDM.AMLUTDLY statement 145  
 NLDM.AUTHDOM statement 146  
 NLDM.AUTHORIZ statement 147  
 NLDM.CDTIME statement 149  
 NLDM.DRDELAY statement 150  
 NLDM.DSRBO statement 150  
 NLDM.ERCOUNT statement 151  
 NLDM.FCTIME statement 151  
 NLDM.KEEPDISC statement 152  
 NLDM.KEEPMEM statement 152  
 NLDM.KEEPPIU statement 153  
 NLDM.KEEPRTM statement 153  
 NLDM.KEEPSESS statement 154  
 NLDM.LOG statement 154  
 NLDM.LUCOUNT statement 155  
 NLDM.MACRF statement 156  
 NLDM.MAXEND statement 157  
 NLDM.NETID statement 157  
   NLDM.PDDNM statement 158  
   NLDM.PERFMEM statement 158  
   NLDM.PEXLSTxx statement 159  
   NLDM.PIUTNUM statement 160  
   NLDM.PIUTSIZE statement 160  
   NLDM.RETRY statement 161  
   NLDM.RTDASD statement 161  
   NLDM.RTM statement 162  
   NLDM.RTMDISP statement 162  
   NLDM.SAW statement 163  
   NLDM.SAWNAM statement 163  
   NLDM.SAWSIZE statement 164  
   NLDM.SDDNM statement 164  
   NLDM.SESSIONS statement 165  
   NLDM.SESSTATS statement 165  
   NLDM.TRACEGW statement 166  
   NLDM.TRACELU statement 166  
   NLDM.TRACESC statement 167  
   NMCSTATUS 353  
   NMCstatus.errorDSN statement 168  
   NOACTY keyword, STATOPT statement 394  
   node  
     focal point 314  
     reactivation 370  
     types, one-character 326  
     VTAM 391  
   NODM.CDRMDEF statement 148  
   NOMONIT keyword, STATOPT statement 394  
   normal messages, command facility panel 367, 369  
   NORMAL statement 367  
   NORMQMAX statement 369  
   NOSTART statement 523  
   notation  
     environment variables xxi  
     path names xxi  
     typeface xxi  
   NOTIFY control file entry for AON notifications 465  
   NPDA.ALCACHE  
     statement 168  
   NPDA.ALERTFWD statement 170  
   NPDA.ALERTLOG  
     statement 172  
   NPDA.ALRTINFP statement 173  
   NPDA.ALTER statement 174  
   NPDA.AUTORATE statement 175  
   NPDA.DSRBO statement 176  
   NPDA.DSRBU statement 177  
   NPDA.ERR\_RATE statement 177  
   NPDA.MACRF statement 178  
   NPDA.MDSIND statement 179  
   NPDA.PDDNM statement 179  
   NPDA.PDFILTER statement 180  
   NPDA.PRELOAD\_BER statement 180  
   NPDA.R (Ratio) statement 181  
   NPDA.RATE statement 182  
   NPDA.SDDNM statement 184  
   NPDA.TECROUTE statement 185  
   NPDA.W (Wrap) statement 185  
   NQNSUP.USE statement 188  
   NRM  
     agent 196  
     CMODE 189  
     CONNINT 190  
     CONNRETRY 190  
     domains 196  
     exclusion lists 197  
     HBRETRY 191

## NRM (*continued*)

HOSTDEST 191  
IPV6SRC 192  
manager 196  
PORT 193  
RESET 194  
SAMPLERATE 194  
status 195  
tasks 198  
NRM.CMODE statement 189  
NRM.CONNINT statement 190  
NRM.CONNRETRY statement 190  
NRM.HBRETRY statement 191  
NRM.HOSTDEST statement 191  
NRM.IPV6SRC statement 192  
NRM.PORT statement 193  
NRM.RESET statement 194  
NRM.SAMPLERATE statement 194  
NRM.STATUS statement 195  
NRM.TYPE statement 196  
NRM.XCLDOM statement 196  
NRM.XCLSRC statement 197  
NRM.XCLTASKN statement 198  
NRM.XCLTASKT statement 198  
NTFYOP 466  
NV-UNIQUE keyword, ALERTFWD statement 171  
NVHOST statement 537  
NVLOCALE statement 538  
NVPORT statement 538  
NVROLE statement 199  
NVSP.srvrname.NUMTHRDS statement 200  
NVSP.srvrname.PDS statement 200  
NVSP.srvrname.PORT statement 201  
NVSP.srvrname.SECURE statement 202  
NVSP.srvrname.TRC statement 203  
NVSP.srvrname.WAIT statement 204

## O

O MONIT statement 370  
O RESET statement 370  
O SECSTAT statement 371  
O SENDMSG statement 372  
object independent method, time interval 569  
OBJPCT keyword, PCLASS statement 378  
OBJTIME keyword, PCLASS statement 378  
obtaining additional status information 388  
OMIT keyword, STATOPT statement 393  
one-character node types, table 326  
online publications  
    accessing xviii  
OPCLASS statement 372  
OpDsPrefix statement 204  
operations management 313, 316  
operator  
    authority, defining 295  
    autotask 296  
    cross-domain sessions 223, 317  
    identifier 373  
    identifier, command facility panel 401  
    profile 317, 383  
    routing messages to 295  
    span of control 238  
        NetView 295  
        VTAM 390  
task, define to RODM access and control task 395  
TSO user ID, surrogate 242

operator group names 26  
OPERATOR statement 373  
OPERIF task message queue 578  
OPERPARM segment of SAF product 134  
OperRecvBrdcst statement 136  
OPKT.STORAGE statement 205  
ORIGNET statement 374  
OSA packet trace management function  
    starting 118  
OUTPUT statement 524

## P

pacing interval  
    enterprise master switch 86  
packet trace data  
    enabling collection 265  
PAGENT started task 250  
PARMSYN keyword  
    CMDDEF statement 301  
PARMSYN statement 375  
PARSE keyword  
    CMDDEF statement 301  
PARTNER statement 375  
PASS keyword, RMTSEC statement 387  
PassTicket 236  
PASSWORD keyword  
    OPERATOR statement 373  
    VPDINIT statement 406  
path names, notation xxi  
PBSDN keyword, DSTINIT statement 320  
PCCU macro 391  
PCLASS keyword, MAPSESS statement 348  
PCLASS statement 376  
PDDNM keyword, DSTINIT statement 320  
PDFILTER command 180  
PDS members  
    reduce I/O rates 127  
PDU38 588  
percentage of command facility panel for messages 329  
performance class, mapping sessions into 348, 376  
PERSIST keyword  
    DSTINIT statement 323  
    PARTNER statement 376  
persistent sessions  
    data transfer 125  
    default 323  
    high performance sessions 375  
PIPE KEEP stage, ENDCMD 83  
PIU trace 160  
PIU trace data  
    concurrent requests 157  
    discarded records 152  
    record retention 153  
    recording 150  
PKTS.STORAGE statement 206  
PLI\_ISA statement 554  
Policy Definitions 353  
POLICY statement 206  
PORT statement 379  
PortNumber statement 525  
PPASS keyword, DSTINIT statement 321  
PPI receiver 129, 185  
PPI statement 525  
PRI keyword  
    KCLASS statement 340  
    MAPSESS statement 348

PRIMARY keyword, DEFFOCPT statement 316  
 PRIMARY\_HEAP\_SIZE statement 554  
 PRINTPDU38 statement 588  
 probe statistics report 106, 107, 108  
 probe summary statistics log file 99  
 probe summary statistics, clearing 96, 98, 99  
 probeid 105  
 processing, ending 324  
 PROFILE statement  
     CONSNAME keyword 383  
     HCL keyword 382  
     IC keyword 383  
     overview 382  
 PROFILEN statement 383  
 program-to-program interface command receiver,  
     defining 309  
 PROTOCOL statement 384  
 publications  
     accessing online xviii  
     NetView for z/OS xv  
     ordering xviii  
 PURGE command 313  
     exception list 159  
 PURGETIMER statement 539  
 PWD statement 207

## Q

QUELIMIT statement 539  
 queue  
     Dbserver task message 573  
     EVENTMGR task message 574  
     IPC task message 574  
     IRMGR task message 575  
     MAINTASK task message 576  
     NETCMD task message 577  
     NETCON task message 577  
     OPERIF task message 578  
     RCMGR task message 579  
     Resource Trait subtask 580  
     VIEWMGR task message 580  
     VSTATMGR task message 581  
 QUIES\_WAIT\_TASK\_TIME statement 555  
 QUIES\_WAIT\_USER\_TIME statement 556

## R

RACF security  
     OPERPARM segment 134  
 ratio, error-to-traffic (E/T) 181  
 RCM  
     RCMRODMUser 208  
 RCMGR task message queue 579  
 RCMRODMUser statement 208  
 receiving network status updates 371  
 RECOVERY control file entry 469  
 regulation event, TCP traffic 113  
 regulation event, UDP traffic 114, 115  
 remote  
     communications 214, 220  
     operator, authorize or block 386  
 RENAME keyword, KCLASS statement 339  
 REP statement 384  
 report  
     IDS probe statistics 106, 107, 108  
 report record logging 183

REPORTS statement 183  
 RES\_CORR keyword, W (Wrap) statement 187  
 RESIDENT control file entries in the control file 474  
 RESIDENT keyword  
     CMDDEF statement 301  
 resource  
     activation by status monitor 346  
 resource discovery 61, 225  
 Resource Trait subtask 580  
 resource traits change method, time interval 571  
 resources  
     resource discovery 61  
 response time monitor measurement keywords 376  
 RESTORE TIMER command  
     starting 120  
 RESTORE.CGLOBAL statement 208  
 RESUME keyword, LOGINIT statement 344  
 RETRY keyword  
     DSTINIT statement 323  
     NLDM.CDRMDEF statement 148  
 RETRY statement 386  
 RetryInterval statement 526  
 RETRYTIME statement 540  
 revision codes xx  
 REXEC server 209, 211  
 REXEC server files  
     definition statement  
         PROTOCOL 384  
 REXEC.PORT statement 209  
 REXEC.SOCKETS statement 210  
 REXEC.TCPNAME statement 211  
 REXX.CMDENV.name statement 211  
 REXX.FUNCPKGLIST.LCL.name statement 212  
 REXX.FUNCPKGLIST.SYS.name statement 213  
 REXX.FUNCPKGLIST.USR.name statement 213  
 RMTALIAS statement 214  
 RMTCMD processing  
     security 217  
 RMTINIT.IP statement 216  
 RMTINIT.PORT statement 216  
 RMTINIT.SARefresh statement 217  
 RMTINIT.SECOPT statement 217  
 RMTINIT.SNA statement 218  
 RMTINIT.SOCKETS statement 218  
 RMTINIT.TCPNAME statement 219  
 RMTOP keyword, RMTSEC statement 387  
 RMTSEC statement 386  
 RMTSYN statement 220  
 RODM  
     access and control task 384  
     automation task definition statement  
         CMDRCVR 309  
         REP 384  
         TASK 395  
     definition statement 541  
         ASYNC\_TASKS 541  
         CELL\_POOLS 541  
         CHARACTER\_VALIDATION 543  
         CHECKPOINT\_FUNCTION 544  
         CONCURRENT\_USERS 545  
         CONNECT\_VIOLATION\_MESSAGE statement 546  
         DUMP\_FOR\_BAD\_USER\_DATA 546  
         DUMP\_LIMIT 547  
         DUMP\_SCOPE 548  
         EXTEND\_HEAP\_SIZE 549  
         IO\_QUEUE\_THRESHOLD 549  
         LOG\_LEVEL 550

RODM (*continued*)

- definition statement (*continued*)
  - MAX\_CHUNK 550
  - MAX\_SEGMENT\_NUM 551
  - MAX\_WINDOW\_NUM 551
  - MLOG\_LEVEL 552
  - MTRACE\_TYPE 553
  - PLI\_ISA 554
  - PRIMARY\_HEAP\_SIZE 554
  - QUIES\_WAIT\_TASK\_TIME 555
  - QUIES\_WAIT\_USER\_TIME 556
  - SEC\_CLASS 556
  - SEC\_RNAME 558
  - SEGMENT\_POCKETS 559
  - SSB\_CHAIN 560
  - TRANSPARENT\_CHECKPOINT 560
  - WAIT\_ALLOC\_TIME 561
  - WINDOW\_CHKPT\_TIME 562
  - WINDOW\_POCKETS 562
  - distribute commands 395
- RODMID statement 590
- RODMname statement 222
- RODMNAME statement 590
- ROUTECDE statement 222
- routing, command 239
- ROWSA
  - TCPCONN 260
- ROWSD
  - TCPCONN 261
  - RRD statement 223
  - RRS interfaces 58
  - RSH server definitions 224
  - RSH server files
    - definition statement
      - PROTOCOL 384
  - RSH.PORT statement 224
  - RSH.SOCKETS statement 224
- RTDEF keyword, PCLASS statement 377
- RTM data 162
  - performance class definitions 158
  - recording 150
- RTMDATA keyword, KCLASS statement 339
- RTNDEF.BASE.COLLECTOR.APPL 225
- RTNDEF.BASE.COLLECTOR.STACK 225
- RTNDEF.BASE.COLLECTOR.SYSTEM 226
- RTNDEF.BASE.HB.APPL 226
- RTNDEF.BASE.MEMBER 226
- RTNDEF.BASE.PROCESSOR.APPL 227
- RTNDEF.BASE.PROCESSOR.STACK 227
- RTNDEF.BASE.PROCESSOR.SYSTEM 228
- RTNDEF.BASE.PROVIDER 228
- RTNDEF.COLLECTOR.APPL.NETVIEW 228
- RTNDEF.PROCESSOR.STACK 229
- RTT.PORT statement 229
- RTT.SOCKETS statement 230
- RTT.TCPNAME statement 230
- RULEFILE statement 540
- running AON and SA/390 in the same NetView domain 439

**S**

- SAF calls, tracing 270, 271
- SAFrefresh keyword, RMTINIT statement 217, 219
- Sample Policy Definitions 353
- samples, location in NetView program 1
- SAW keyword
  - KCLASS statement 340
- SBSDN keyword, DSTINIT statement 321
- scan detection event 110
- scan detection event, IDS 109
- screen format statements
  - %INCLUDE statement 333
  - ACTION statement 292
  - CMDLINE statement 308
  - COLUMNHEAD statement 310
  - HELD statement 327
  - HOLDPCNT statement 329
  - HOLDWARN statement 330
  - IMDAREA statement 331
  - INDENT statement 335
  - LASTLINE statement 341
  - LOCKIND statement 343
  - MLINDENT statement 349
  - NOPREFIX statement 367
  - NORMAL statement 367
  - NORMQMAX statement 369
  - PREFIX statement 379
  - TITLE statement 396
  - TITLEDATE statement 398
  - TITLEDOMID statement 399
  - TITLEOPID statement 401
  - TITLESTAT statement 402
  - TITLETIME statement 404
- SDDNM keyword, DSTINIT statement 321
- SEC keyword
  - KCLASS statement 340
  - MAPSESS statement 348
- SEC\_CLASS statement 556
- SEC\_RNAME statement 558
- secondary network resource status monitor 371
- SECOPTS.AUTHCHK statement 231
- SECOPTS.CMDAUTH statement 231
- SECOPTS.OPERSEC statement 234
- SECOPTS.OPSPAN statement 238
- SECOPTS.RMTAUTH statement 239
- SECOPTS.SARESAUTH statement 240
- SECOPTS.SPANAUTH statement 241
- SECOPTS.SPANCHK statement 241
- SECOPTS.SURROGAT statement 242
- SECOPTS.WEBAUTH statement 242
- security
  - command authority 231
  - NACMD.OPID statement 140
  - operator 234
  - resource level 240
  - span of control 241
- security checking 58
- SECURITY keyword
  - CMDDEF statement 301
- SEGMENT\_POCKETS statement 559
- SENDMSG statement 388
- ServerFallback statement 527
- ServerLocation statement 528
- ServerPort statement 529
- service xix
  - service management connect xix
- service, class of 312
- SESS command 165
- SESSFAIL keyword, KCLASS statement 339
- session
  - CNM data transfer 126
  - LU, authorization 126
  - monitor
    - conversations 148, 323

session (*continued*)
   
     monitor (*continued*)
   
         session data retrieval 148
   
     parallel 126
   
     persistent 125
   
     TAF 324
   
**SESSION (SNA) statement** for sessions monitored by AON/SNA 475
   
 session awareness data 163, 164
   
 session configuration data
     authorization 147
   
 session data 139
   
 session monitor
     availability data 154, 165
     cross-domain session 149
     DASD session wrapping 154
     enabling 265
     ER data 161
     explicit routes 151
     external log 154
     flow control data request 151
     gateway tracing 166
     initial conversation 161
     keep class 152
     logical units, number 155
     LU tracing 166
     PIU trace data 150, 152, 153, 157
     recording a session to database 150, 151
     response time collection period 153
     RTM data 150, 162
     SSCP tracing 167
   
**SESSNORM keyword**, KCLASS statement 339
   
**SMC** xix
   
**SMF log** 154
   
**SNA keyword**, RMTSYN statement 221
   
**SNA Topology Manager**
     response to GMFHS 585
   
**SNA-MDS keyword**, .ALERTFWD statement 171
   
**SNAPRQ keyword**, VPDINIT statement 406
   
**SNMP trap automation** 38
   
**SOCKETS statement** 389
   
**span definitions** 241
   
**span of control** 238
     authorization 241
     span of control 336
   
**SPAN statement**
     definition 390
   
**SPASS keyword**, DSTINIT statement 321
   
**SRATIO command** 181, 185
   
**SSB\_CHAIN statement** 560
   
**SSCP**
     cross-domain, name 148
     session awareness data for 337
     tracing 167
   
**SSI.PPI statement** 243
   
**SSI.ProcString statement** 243
   
**SSI.ReviseTable statement** 244
   
**ST keyword**, W (Wrap) statement 187
   
**stack information** 225
   
**START command** 252
   
**START SPAN command** 238, 390
   
**START TASK command** 223
   
**statements**
     stylevar 246
   
**statements**, location in samples 1
   
**statistic record recording** 173
   
**statistical records**
     distributed DVIPA 79
   
**STATOPT**
     default descriptions 393
     statement 391
   
**status**
     indicators, command facility panel 402
   
**status monitor**
     as secondary resource 371
     displaying commands or command lists 300
     functions 391
     initialization message for resources 372
     MONIT function 346
     reactivating nodes 370
   
**STOP command** 336
   
**STORAGE statement** 591
   
**styleMsg statement** 245
   
**styleMsg.wrap statement** 245
   
**stylevar statement** 246
   
**SUBSYSTEM control file entry** for NVAS subsystem interface
     with SNA help desk 475
   
**subsystem interface statement** 243
   
**SUBSYSTEM statement** 395
   
**subtask, Resource Trait** 580
   
**SuppChar statement** 247
   
**support** xix
   
**suppression character** 247
   
**SWITCH command** 313
   
**switching traffic load back to the primary line** 481
   
**SWRAP command** 181, 185
   
**synonyms, defining**
     command lists 301
     commands 301
     keywords 301
   
**syntax conventions** 4
   
**syslog server** 121
   
**sysplex**
     entry point 87
   
**sysplex master**
     connectivity checks 84
   
**system authorization facility product**
     APPL class 235
   
**System Automation for z/OS** 265

## T

**TAF sessions** 324
   
**TALRTCFG statement** 530
   
**TAMEL.CONV.sysdef statement** 248
   
**TAMEL.PORT statement** 249
   
**TAMEL.SOCKETS statement** 249
   
**TAMEL.TCPNAME statement** 250
   
**TAMEL.TTLS statement** 250
   
**TAMEL.USETCPPIP statement** 251
   
**TARGOP keyword**, RMTSEC statement 387
   
**task groups, web application** 276
   
**TASK statement** 591
   
**TASK.taskname.INIT statement** 252
   
**TASK.taskname.MEM statement** 253
   
**TASK.taskname.MOD statement** 253
   
**TASK.taskname.PRI statement** 254
   
**tasks**
     reserved, web application 277
     web application 277
   
**tasks, defining** 253
   
**tasks, priority** 254
   
**tasks, tracing** 272

TCP traffic regulation event 113  
 TCP/IP  
     connectivity  
         CNMTAMEL 248  
         DSIIPLOG 121  
         DSIREXCF 211  
         DSIRSH 224  
         DSIRTTR 229  
         DSIWBTSK 275  
 TCP/IP connection data  
     enabling collection 265  
 TCP/IP connection management  
     starting 119  
 TCP/IP feature 263  
 TCP/IP stack interfaces 60  
 TCP390 control file entry 476  
 TCPANAME statement 395  
 TCPCONN.DASD statement 255  
 TCPCONN.DSRBO statement 256  
 TCPCONN.ERRMSG statement 256  
 TCPCONN.GTF.stackname statement 257  
 TCPCONN.HASHSIZE.stackname statement 257  
 TCPCONN.KEEP.stackname statement 258  
 TCPCONN.MACRF statement 259  
 TCPCONN.PDDNM statement 259  
 TCPCONN.ROWSA.stackname statement 260  
 TCPCONN.ROWSD.stackname statement 261  
 TCPCONN.SDDNM statement 261  
 TCPIP connection data 138  
 TCPIP connection data, inactive 137  
 TCPIPCOLLECT 265  
     PKTS subtower 265  
     TCPConn subtower 265  
 TCPname statement 262  
 TCPserver statement 262  
 telnet active listeners 60  
 TEMA  
     CONINACT subtower 265  
     CONNACT subtower 265  
     DVCONN subtower 265  
     DVDEF subtower 265  
     DVROUT subtower 265  
     DVTAD subtower 266  
     HEALTH subtower 266  
     SESSACT subtower 266  
     SYSPLEX subtower 266  
     TELNET subtower 266  
 TestMode statement 531  
 TGSWITCH control file entry 481  
 threshold  
     IDS 95  
 THRESHOLDS, setting 482  
 time-out value 149  
 TIMER control file entries 486  
 title line  
     command facility panel 396  
     date, command facility panel 398  
     time, command facility panel 404  
 TITLE statement 396  
 TITLEDATE statement 398  
 TITLEDOMID statement 399  
 TITLEOPID statement 401  
 TITLESTAT statement 402  
 TITLETIME statement 404  
 Tivoli  
     training, technical xix  
     user groups xix  
 Tivoli Enterprise Monitoring Server  
     NACMD.OPID statement 140  
 Tivoli NetView for z/OS Enterprise Management Agent 265  
 Tivoli Software Information Center xviii  
 TOWER statement 263  
 TP keyword, MAPSESS statement 349  
 TRACDATA keyword, KCLASS statement 339  
 trace  
     VSAM database performance, warning 592  
 trace data  
     authorization 147  
 TRACE initialization parameters 266, 269  
 TRACE statement 531, 592  
 TRACE.MODE statement 266  
 TRACE.MODFILT statement 267  
 TRACE.MONOPER statement 268  
 TRACE.OPTION statement 269  
 TRACE.SAFopt statement 270  
 TRACE.SAFrc statement 271  
 TRACE.SIZE statement 271  
 TRACE.TASK statement 272  
 TRACEBYTES statement 593  
 TRACEPAGES statement 594  
 traffic regulation event, TCP 113  
 traffic regulation event, UDP 114, 115  
 training, Tivoli technical xix  
 translation  
     alias name 293  
     character 273  
     messages 272  
 transMember statement 272  
 TRANSPARENT\_CHECKPOINT statement 560  
 transTbl statement 273  
 TruncateSV31s statement 533  
 TSO user ID  
     surrogate 242  
 TSOSEERV 489  
 TYPE keyword  
     CMDDDEF statement 301  
     DEFFOCPT statement 314, 315  
 TYPE statement 594  
 typeface conventions xx

## U

UDP traffic regulation event 114, 115  
 UNIX System Services file 111  
 UNSOL keyword, DSTINIT statement 320  
 unsolicited messages, routing to an operator 295  
 UsePortMapper statement 534  
 user groups  
     NetView, on Yahoo xx  
     Tivoli xix  
 UserStatus field 569

## V

VALCLASS statement 405  
 variables, notation for xxi  
 VBUILD macro 391  
 view  
     updates 571  
 VIEWMGR task message queue 580  
 VIPA routes 76  
 vital product data, logging 406  
 VPDINIT statement 406

VPDREQ keyword, VPDINIT statement 406  
VPDSTOR keyword, VPDINIT statement 407  
VPDTASK subtask 406  
VPDWAIT keyword, VPDINIT statement 406  
VR keyword, MAPSESS statement 349  
VSAM  
  database trace, warning 592  
  file, mapping sessions to 339  
  services 320  
VSTATMGR task message queue 581  
VTAM  
  CDRSC definition statement 345  
  spans 390  
VTAMCP statement 273

## W

WAIT\_ALLOC\_TIME statement 561  
WBSORDER statement 274  
web access 274, 275, 278  
web application  
  portfolio definition 276  
  reserved tasks 277  
  tasks 277  
web server, authorization 242  
Web Services Gateway function  
  enabling 265  
Web Services server  
  command response 204  
  enabling 265  
  port 201  
  server initialization files 200  
  SSL encryption 202  
  threads 200  
  trace levels 203  
WEB statement 275  
WEB\_SERVER statement 407  
webmenu statement 276  
WEBSITE statement 278  
WINDOW statement 408  
WINDOW\_CHKPT\_TIME statement 562  
WINDOW\_POCKETS statement 562  
WINDOW.EXEC statement 279  
WLM support 279  
WLM.SubSystemName statement 279  
workload management 279  
wrap count  
  initial 185  
WRAPCNT keyword, NPDA.ALCACHE statement 168

## X

X25MONIT control file entry 490  
XCF.GROUPNUM statement 279  
XCF.MASTDVIPA statement 280  
XCF.PROCSTRnn statement 281  
XCF.RANK statement 282  
XCF.TAKEOVER.CLIST statement 282  
XCF.TAKEOVER.CONVIPnn statement 283  
XCF.TAKEOVER.CONVSNA statement 284  
XCF.TAKEOVER.DELAY statement 284  
XCF.TAKEOVER.DURATION statement 285  
XCF.TAKEOVER.INITWAIT statement 286  
XCF.TAKEOVER.NETCONVS statement 286  
XITBN keyword, DSTINIT statement 321  
XITBO keyword, DSTINIT statement 322

XITCI keyword, DSTINIT statement 322  
XITCO keyword, DSTINIT statement 322  
XITDI keyword, DSTINIT statement 322  
XITVI keyword, DSTINIT statement 322  
XITVN keyword, DSTINIT statement 322  
XITVO keyword, DSTINIT statement 322  
XITXL keyword, DSTINIT statement 322  
XLO keyword, REPORTS statement 184

## Y

Yahoo user group, NetView xx





**IBM**<sup>®</sup>

Printed in USA

SC27-2869-06

