OS/390 SecureWay Communications Server

SNA Resource Definition Reference

Version 2 Release 8

SC31-8565-03

OS/390 SecureWay Communications Server

SNA Resource Definition Reference

Version 2 Release 8

SC31-8565-03

Note:
Before using this information and the product it supports, be sure to read the general information under “Appendix E. Notices” on page 843.

Fourth Edition (September 1999)

This edition applies to OS/390 V2R8 (Program Number 5647-A01).

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):
1-800-227-5088

Internet e-mail:
usib2hpd@vnet.ibm.com

World Wide Web:
http://www.s390.ibm.com/os390

IBMLink:
CIBMORCF at RALVM17

IBM Mail Exchange:
USIB2HPD at IBMMAIL

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1999. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures ix

Tables xi

About This Book xv
Who Should Use This Book xv
How to Use This Book xv
How This Book Is Organized xv

Summary of Changes xvii

Chapter 1. Introduction 1
About This Chapter 1
Major Nodes 2
Adjacent Control Point Major Node 2
Application Program Major Node 2
Channel-Attachment Major Node 2
Cross-Domain Resource Major Node 2
Cross-Domain Resource Manager Major Node 2
External Communication Adapter Major Node 2
Local Non-SNA Major Node 2
Local SNA Major Node 3
LU Group Major Node 3
Model Major Node 3
NCP Major Node 3
Switched Major Node 3
TCP/IP Major Node 3
Transport Resource List Major Node 4
Routing Functions and Dynamic Reconfiguration 4
Adjacent SSCP Tables 4
Path Definition Statements 4
Dynamic Reconfiguration and Dynamic Change 5
Transmission Group Profile 5
Network Node Server List 5
User-Defined Tables and Session Awareness Data Filter 6
Associated LU Table 6
Class of Service 6
CMIP Services Directory Definition File. 6
Interpret Table 7
Logon Mode Table 7
Message-Flooding Prevention Table 7
Model Name Table 8
Session Awareness (SAW) Data Filter 8
Unformatted System Services (USS) Table 8
Coding VTAM Definitions 8
Format of Definition Statements 9
Format of Macroinstructions 10
Rules for Coding Continued Lines 12
Rules for Coding Comments 12
Using a Default 13
Using the Sift-Down Effect in VTAM Definition Statements 13
Restrictions on Names 14

© Copyright IBM Corp. 1984, 1999

Using MVS System Symbols in VTAM Definition Statements and Macroinstructions . . . 14
Restrictions on Use of Assembler Features . . . 15
How to Read Railroad Track Syntax Diagrams . . . 15
Filing Definition Statements . . . 18
Filing Macroinstructions . . . 18

Chapter 2. Major Nodes . . . 19
Major Nodes . . . 19
About This Chapter . . . 19
Adjacent Control Point Major Node . . . 19
Coding Notes . . . 20
Full Syntax . . . 20
Operand Descriptions . . . 21
Application Program Major Node . . . 25
Registering Application Programs . . . 25
Non-LU 6.2 Application Programs . . . 25
LU 6.2 Application Programs . . . 27
Coding Notes . . . 28
Full Syntax . . . 29
Operand Descriptions . . . 33
Channel-Attachment Major Node . . . 65
Channel-to-Channel Adapter Support . . . 65
Multipath Channel Support . . . 66
Channel-Attached NCP Support . . . 67
Coding Notes . . . 70
Full Syntax . . . 70
Operand Descriptions . . . 71
CHANCON . . . 72
CONFGDS . . . 73
CONFGPW . . . 73
DELAY . . . 73
ISTATUS . . . 74
LNCTL . . . 74
MAXBFRU . . . 75
MAXDATA . . . 76
MIH . . . 76
name . . . 77
PUTYPE . . . 78
READ . . . 78
REPLYTO . . . 79
SPAN . . . 79
TGN . . . 80
TYPE . . . 80
WRITE . . . 80
Cross-Domain Resource (CDRSC) Major Node . . . 81
Independent LU . . . 81
APPN Cross-Domain Resources . . . 82
Rules for Multiple Definition of Resources . . . 83
Coding Notes . . . 93
Full Syntax . . . 93
Operand Descriptions . . . 95
Cross-Domain Resource Manager (CDRM) Major Node . . . 113
Coding Notes . . . 115
Full Syntax . . . 116
Operand Descriptions . . . 118

External Communication Adapter (XCA) Major Node . . . 133
LAN and ATM LAN Emulation Connections . . . 133
Enterprise Extender Connections . . . 136
ATM Native Connections . . . 138
Coding Notes . . . 143
Full Syntax . . . 143
Operand Descriptions . . . 154
Local Non-SNA Major Node . . . 191
Coding Notes . . . 192
Full Syntax . . . 192
Operand Descriptions . . . 193
Local SNA Major Node . . . 202
Coding Notes . . . 204
Full Syntax . . . 205
Operand Descriptions . . . 210
LU Group Major Node . . . 240
Coding Notes . . . 241
Full Syntax . . . 241
Operand Descriptions . . . 243
Model Major Node . . . 253
Coding Notes . . . 255
Full Syntax . . . 256
Operand Descriptions . . . 261
Network Control Program (NCP) Major Node . . . 295
Identifying the Communication Controller . . . 296
NCP Definition Statements for the NCP . . . 297
NCP Definition Statements for Resources Controlled by the NCP . . . 300
Coding Notes . . . 306
Full Syntax . . . 308
Operand Descriptions . . . 321
Switched Major Node . . . 392
PU and PATH Definition Statements for a Switched Subarea. . . . 393
PU and PATH Definition Statements for the XCA Major Node . . . 393
Coding Notes . . . 396
Enterprise Extender Connections . . . 396
XCA and NCP Connections . . . 399
XCA and NCP Connections . . . 401
Full Syntax . . . 401
Enterprise Extender Connections . . . 408
Local (Source) and Remote (Destination) Service Access Point Detemination 410
Operand Descriptions . . . 410
Transport Resource List Major Node . . . 469
Coding Notes . . . 470
Full Syntax . . . 470
Operand Descriptions . . . 472

Chapter 3. Routing and Dynamic Reconfiguration . . . 481
Routing and Dynamic Reconfiguration . . . 481
About This Chapter . . . 481
VTAM Routes . . . 481
Dynamic Path Update . . . 485
Dynamic Reconfiguration and Dynamic Change . . . 494
Coding Notes . . . 497
Full Syntax . . . 497
Operand Descriptions . . . 498
Adjacent SSCP Table . . . 501

Coding Notes 501
Full Syntax 503
Operand Descriptions 504
Adjacent Cluster Routing Definitions. 508
Coding Notes 509
Full Syntax 510
Operand Descriptions 510
APPN Transmission Group Profile 513
TGP Definition Statement 513
IBM-Supplied Transmission Group Profiles 513
Full Syntax 514
Operand Descriptions 515
APPN-to-Subarea Class-of-Service Mapping Table 521
Coding Notes 522
Full Syntax 522
Operand Descriptions 522
Subarea-to-APPN Class-of-Service Mapping Table 524
Coding Notes 524
Full Syntax 525
Operand Descriptions 525
Border Node COS Mapping Definitions. 526
Coding Notes 527
Full Syntax 528
Operand Descriptions 528
Network Node Server List 529
Coding Notes 530
Full Syntax 530
Operand Descriptions 531

Chapter 4. Start Options 533
Start Options 533
About This Chapter 533
Coding VTAM Start Options 533
Summary of Start Options 549
Descriptions of Start Options 560

Chapter 5. User-Defined Tables and Data Filter 661
User-Defined Tables and Data Filter. 661
About This Chapter 661
Associated LU Table 661
Class of Service (COS) 664
CMIP Services Directory Definition File. 681
Interpret Table 690
Coding Notes 691
Coding Logon-Interpret Routines 692
Installing and Changing Interpret Tables 693
Full Syntax 694
Operand Descriptions 695
Logon Mode Table 698
Coding Notes 699
Installing a Logon Mode Table 700
Full Syntax 700
Operand Descriptions 701
Message-Flooding Prevention Table 714
Coding Notes 714
Installing a Message-Flooding Prevention Table 714

Considerations for Message-Flooding Prevention . . . 715
Example of Message Suppression . . . 715
Full Syntax . . . 716
Operand Descriptions . . . 717
Model Name Table . . . 720
Coding Notes . . . 721
Installing a Model Name Table . . . 721
Full Syntax . . . 721
Operand Descriptions . . . 722
Subarea Mapping Table . . . 723
Coding Notes . . . 723
Full Syntax . . . 724
Operand Descriptions . . . 724
Session Awareness (SAW) Data Filter . . . 725
Coding Notes . . . 726
Full Syntax . . . 727
Operand Descriptions . . . 727
SAW Sense Filter . . . 729
Coding Notes . . . 730
Full Syntax . . . 730
Operand Descriptions . . . 730
Unformatted System Services (USS) Tables . . . 731
Customizing USS Tables . . . 732
Order of USS Table Use . . . 733
Installing a User-Defined USS Table . . . 735
Coding a User-Defined USS Table . . . 735
Coding Notes . . . 736
Full Syntax . . . 737
Operand Descriptions . . . 739
Customizing Messages . . . 753
Defining VTAM Operator Commands . . . 756
Defining Terminal Operator Commands . . . 757
LOGON Command . . . 758
Full Syntax . . . 758
Operand Descriptions . . . 759
LOGOFF Command . . . 762
Full Syntax . . . 762
Operand Descriptions . . . 762
IBMTEST Command . . . 765
Full Syntax . . . 765
Operand Descriptions . . . 766

Appendix A. IBM-Supplied Tables . . . 769
Default Logon Mode Table (ISTINCLM) . . . 769
Session-Level USS Table (ISTINCDT) . . . 784
Operation-Level USS Table (ISTINCNO) . . . 785
APPN Class-of-Service (COS) Definitions (COSAPPN) . . . 798
COSAPPN . . . 798
ISTACST2 . . . 810
APPN Transmission Group (TG) Profile Definitions (IBMTGPS) . . . 824
Ethernet . . . 824
Token Ring . . . 824
ISDN Non-Switched . . . 824
ISDN Switched . . . 824
SDLC Non-Switched . . . 825
SDLC Switched . . . 825

X.25 825
16M Token Ring 825
Channel 825
ESCON Channel 825
ATM TGs over PVCs (Campus [Private] ATM Best Effort) 825
ATM TGs over PVCs (Campus ATM Reserved Bandwidth) 825
ATM TGs over PVCs (WAN [Public] ATM Best Effort) 825
ATM TGs over PVCs (WAN ATM Reserved Bandwidth) 825
ATM TGs over SVCs (Campus ATM Best Effort) 825
ATM TGs over SVCs (Campus ATM Reserved Bandwidth) 825
ATM TGs over SVCs (WAN ATM Best Effort) 825
ATM TGs over SVCs (WAN ATM Reserved Bandwidth) 826
Enterprise Extender, Campus 826
Enterprise Extender, Wide Area Network 826
Message Flooding Table (ISTMSFLD) 826
CMIP Services Directory Definition File (ACYDDF) 831

Appendix B. Examples of USS Command Conversion 833

Appendix C. VTAMTOPO Filtering Option Reporting 837

Appendix D. Information Apars 841
SNA Information Apars 841

Appendix E. Notices 843
Programming Interface Information 845
Trademarks 846

Bibliography 849
SecureWay Communications Server for OS/390 Publications 849
Related Publications 849
Softcopy Information 849
Planning 849
Resource Definition, Configuration, and Tuning 850
Operation 850
Customization 850
Writing Application Programs 850
Diagnosis 851
Messages and Codes 852
APPC Application Suite 852
Multiprotocol Transport Networking (MPTN) Architecture Publications 852
Redbooks 852

Index 855

Readers' Comments — We'd Like to Hear from You 875

Figures

1. Adjacent Cluster Routing Definitions 509
2. APPN-to-Subarea COS Mapping Table Example 522
3. Subarea-to-APPN COS Mapping Table Example 524
4. COS Mapping Table Example 527
5. Subarea Mapping Table Example 723
6. USS Message Layout in Storage 739

© Copyright IBM Corp. 1984, 1999

Tables

1. Adjacent Control Point Major Node Definition Statements and Operands 19
2. DYNLU Combinations 22
3. Application Program Major Node (Non-LU 6.2) Definition Statements and Operands. 25
4. Application Program Major Node (LU 6.2) Definition Statements and Operands. 27
5. Channel-Attachment (CTCA) Major Node Definition Statements and Operands. 65
6. Channel-Attachment (MPC) Major Node Definition Statements and Operands. 67
7. Channel-Attachment (NCP) Major Node Definition Statements and Operands. 69
8. Rules for Multiple Definition of Resources 83
9. Examples of How LUALIAS Values Are Merged. 85
10. Resource Types for NQNMODE Merges 87
11. Matrix for NQNMODE Merges, Part 1 88
12. Matrix for NQNMODE Merges, Part 2 89
13. Notes for NQNMODE Merges Tables 90
14. Cross-Domain Resource Major Node Definition Statements and Operands 92
15. How NQNMODE Relates to CDRSCs 109
16. Cross-Domain Resource Manager Major Node Definition Statement and Operands. 113
17. External Communication Adapter Major Node - LAN, ATM LAN Emulation, or Enterprise Extender Connections, Subarea Node 134
18. External Communication Adapter Major Node—LAN, ATM LAN Emulation, or Enterprise Extender Connections, Peripheral Node 137
19. External Communication Adapter Major Node—ATM Native Connection, Port 139
20. External Communication Adapter Major Node—ATM Native Connection, TGs over PVCs 140
21. External Communication Adapter Major Node—ATM Native Connection, TGs over SVCs 141
22. External Communication Adapter Major Node - ATM Native Connection, TGs to Connection Networks 142
23. XCA AUTOGEN without User-Supplied Seeds 157
24. XCA AUTOGEN with Single-Character User-Supplied Seeds 157
25. XCA AUTOGEN with 3-Character User-Supplied Seeds. 157
26. Local Non-SNA Major Node Definition Statements and Operands 191
27. Local SNA Major Node Definition Statements and Operands 202
28. LU Group Major Node Definition Statements and Operands 240
29. Model Major Node Definition Statements and Operands 253
30. XCF Model PU Operands Used With TRLE 290
31. NCP Major Node Statements and Operands 301
32. NCP Line Definition Dependency for Coding VTAMTOPO Keyword . . . 390
33. Switched Major Node Definition Statements and Operands 393
34. Transport Resource List Major Node Statements and Operands. 470
35. Definition Statement for Routes 481
36. Definition Statements for Dynamic Update NCP Paths 485
37. Definition Statements for Dynamic Update VTAM Paths. 490
38. Definition Statements for Dynamic Reconfiguration with VARY DRDS (ADD operation) 495
39. Definition Statements for Dynamic Reconfiguration with VARY DRDS (DELETE Operation) 496

© Copyright IBM Corp. 1984, 1999

40. Definition Statements for Dynamic Reconfiguration with VARY DRDS (MOVE Operation) . . . 497
41. Definition Statements for ADJSSCP Table . . . 501
42. Definition Statements for ADJCLUST Table . . . 509
43. Definition Statement for Transmission Group Profile (TGP) . . . 514
44. CAPACITY Values When Displayed . . . 515
45. Definition Statements for APPNTOSA Table . . . 522
46. Definition Statements for SATOAPPN Table . . . 524
47. Definition Statements for COSMAP Table . . . 527
48. Definition Statement for Network Node Server List . . . 530
49. Start Options That Define SSCP or CP Characteristics . . . 549
50. Start Options That Affect VTAM Initialization . . . 550
51. Start Options That Define Storage Usage . . . 551
52. Start Options That Affect Session Control . . . 551
53. Start Options That Affect Connectivity . . . 552
54. Start Options That Affect Session Security . . . 553
55. Start Options That Affect Messages . . . 553
56. Start Options That Affect Recording and Statistics . . . 554
57. Start Options That Affect Traces and Dumps . . . 554
58. Start Options That Define Buffer Pools . . . 555
59. Start Options That Affect Performance and Tuning. . . . 555
60. Start Options That Define APPN Characteristics . . . 557
61. Start Options That Were Zappable Constants . . . 559
62. VTAM Buffer Pools . . . 567
63. IBM-Supplied VTAM Buffer Pool Values . . . 570
64. IOBUF Buffer Size and Number of Buffers per Page . . . 572
65. Node Type Functional Summary . . . 596
66. XNETALS Matrix . . . 657
67. Associated LU Table Macroinstructions and Operands . . . 662
68. Subarea COS Macroinstructions and Operands. . . . 665
69. APPN COS Definition Statements and Operands . . . 668
70. Default Values for LINEROW CAPACITY Operand . . . 670
71. Default Values for LINEROW COSTBYTE Operand . . . 671
72. Default Values for LINEROW COSTTIME Operand . . . 672
73. Default Values for LINEROW PDELAY Operand . . . 674
74. Default Values for LINEROW WEIGHT Operand . . . 678
75. Interpret Table Macroinstructions and Operands . . . 690
76. Logon Mode Table Macroinstructions and Operands . . . 698
77. Exponent and Mantissa Values for Each RU size . . . 711
78. Message-Flooding Prevention Table Macroinstructions and Operands . . . 714
79. Model Name Table Macroinstructions and Operands . . . 721
80. Definition Statements for SAMAP Table. . . . 723
81. Session Awareness Data Filter Macroinstructions and Operands . . . 726
82. SAW Sense Filter Definition Statements and Operands . . . 730
83. Description of the IBM-Supplied USS Tables . . . 732
84. VTAM Messages Search Order for VTAM Messages . . . 734
85. VTAM Messages Search Order for a Program Operator. . . . 734
86. VTAM Commands Search Order for VTAM Commands . . . 734
87. Input Translation Tables Search Order . . . 735
88. USS Table Macroinstructions and Operands . . . 735
89. Valid Character Strings for Message Definition . . . 740
90. USS Messages and VTAM Message Equivalents . . . 754
91. LOGON Command and Operands . . . 758
92. LOGOFF Command and Operands . . . 762
93. IBMTEST Command and Operands . . . 765
94. Connected Switched PU Report . . . 837

95. SNA Information Apars 841

About This Book

This book, *OS/390 SecureWay Communications Server: SNA Resource Definition Reference*, is the fourth edition for the *SecureWay Communications Server for OS/390 V2R8*. It presents the meaning and coding syntax of VTAM® resource definition statements and operands, NCP resource definition statements and operands that affect VTAM, start options, and macroinstructions and operands for user-defined tables.

To plan for the tasks of installing VTAM and defining resources to it, consult *OS/390 SecureWay Communications Server: SNA Migration*.

When you are ready to install VTAM and define resources to it, you can find the broader, more conceptual information about installation and definition in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

This book contains the specifics of resource definition.

Who Should Use This Book

This book is intended to help system programmers define resources to VTAM.

How to Use This Book

You can use this book as a reference for specific operator commands and as background reading to help understand how to operate VTAM.

How This Book Is Organized

This book consists of the following chapters and appendixes:

- Chapter 1, “Introduction”

  This chapter describes the symbols and coding conventions used for the definition statements and macroinstructions described in this book. Descriptions of the sift-down effect and each type of major node are also included.
- Chapter 2, “Major Nodes”

  This chapter describes the VTAM definition statements used to define major nodes. Descriptions of NCP definition statements and their operands are given only when they affect VTAM.
- Chapter 3, “Routing and Dynamic Reconfiguration”

  This chapter describes the VTAM definition statements used, in general, to define routes and to dynamically reconfigure and dynamically change resources.
- Chapter 4, “Start Options”

  This chapter describes VTAM start options and how you can code them to set various conditions that are to exist while VTAM is running.
- Chapter 5, “User-Defined Tables and Data Filter”

  This chapter describes how to create or modify the associated LU table, class-of-service table, CMIP services directory definition file, interpret table, logon mode table, message-flooding prevention table, model name table, and unformatted system services (USS) table. It also describes how to code the session awareness (SAW) data filter.
- Appendix A, “IBM-Supplied Tables”

© Copyright IBM Corp. 1984, 1999

This appendix contains listings of the IBM®-supplied default tables.

- Appendix B, “Examples of USS Command Conversion”

  This appendix contains examples of how to code USS tables to convert commands.
- Appendix C, “VTAMTOPO Filter Option Reporting”

  This appendix summarizes the results of using the VTAMTOPO filtering option for reporting a switched PU under an NCP.
- Appendix D, “Information Apars”

  This appendix contains SNA Information Apars for previous releases of this book.
- Appendix E, “Notices”

  This appendix contains notices and trademarks of the IBM Corporation and of other companies.

Where to Find Related Information on the Internet

You may find the following information helpful.

You can read more about VTAM, TCP/IP, OS/390, and IBM on these Web pages. For up-to-date information about Web addresses, please refer to informational APAR II11334.

Home Page	Web address
IBM SecureWay Communications Server product	http://www.software.ibm.com/network/commserver/
IBM SecureWay Communications Server support	http://www.software.ibm.com/network/commserver/support/
OS/390	http://www.ibm.com/os390/
OS/390 Internet Library	http://www.ibm.com/s390/os390/bkserv/
IBM Systems Center publications	http://www.ibm.com/redbooks
IBM Systems Center flashes	http://www.ibm.com/support/techdocs
VTAM and TCP/IP	http://www.software.ibm.com/network/commserver/about/csos390.html
IBM	http://www.ibm.com

For definitions of the terms and abbreviations used in this book, you can view or download the latest *IBM Networking Softcopy Glossary* at the following Web address:

http://www.networking.ibm.com/nsg/nsgmain.htm

Note: Any pointers in this publication to websites are provided for convenience only and do not in any manner serve as an endorsement of these websites.

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Summary of Changes

Summary of Changes
for SC31-8565-03
SecureWay Communications Server for OS/390 V2R8

This edition contains new and changed information that is indicated by vertical lines in the left margin.

New Information

- MULTGSUP operand to the External Communications Adapter Major Node section
- ENCRTYPE operand to Application Program Major Node, Cross-Domain Resource Major Node, Local SNA Major Node, LU Group Major Node, Model Major Node, NCP Major Node, and Switched Major Node sections
- SLOWVAL command to Start Options

Changed Information

- The term eNetwork™ is replaced by SecureWay® as part of our product name. The new name is SecureWay Communications Server.
- Default on DELAY parameter in the Local SNA Major Node section
- Buffer Pools Start Options table to include CRA4BUF and CRA8BUF values in the Start Options section
- ENCR operand
- HOSTSA, HNTSIZE, NODETYPE, and SACONNS in the Start Options section
- MAXLURU is now modifiable
- VIT Operand diagram in the Start Option section to include SUBTRACE
- PATH statements in Switched Major Node
- Display for an active TRLE entry
- MAPSESS, PRINET, and SECNET operands in Session Awareness Data Filter
- USS Table (ISTINCNO)
- The Bibliography has been revised to show book number dash levels and delivery format.

Summary of Changes
for SC31-8565-02
eNetwork Communications Server for OS/390 V2R7

New Information

- IP option to the FRELAY operand in the NCP Major Node section
- IPRESOLV operand in Routing and Dynamic Reconfiguration
- SAVERSCV start option in the Start Options section
- VIT operand and CIA in the Start Options section
- OSA-Express information to the Transport Resource List major Node section
- SORDER operand to the Adjacent SSCP Table section
- As part of the name change of OpenEdition® to OS/390® UNIX System Services, occurrences of OS/390 OpenEdition have been changed to OS/390 UNIX

© Copyright IBM Corp. 1984, 1999

System Services or its abbreviated name, OS/390 UNIX. OpenEdition may continue to appear in messages, panel text, and other code with OS/390 UNIX System Services.

Changed Information

- PUTYPE=4 in the External Communication Adapter Major Node section
- ENHADDR operand in the Start Options section
- Message information for the FSIRFMSG start option

Deleted Information

NOXNETLS from the NETID syntax diagram in the NCP Major Node section

Summary of Changes
for SC31-8565-01
eNetwork Communications Server for OS/390 V2R6

New Information

- Enterprise Extender network to LAN and ATM LAN connections in the External Communication Adapter Major Node section
- Syntax diagram to the CAPACITY operand for ATM native connections
- IPPORT, IPTOS, LIVTIME, SRQRETRY, and SRQTIME operands to the External Communication Adapter Major Node section
- INCLUD0E operand to the Local SNA Major Node, Model Major Node, NCP Major Node, and Switched Major Node sections
- CLRSESSQ operand to the Model Major Node and Switched Major Node sections
- Enterprise Extender to the Switched Major Node section
- Operands IPADDR, MLTGRPI, SAPADDR, and TGCONF to the Switched Major Node section
- MPCUSAGE operand to Transport Resource List Major Nodes
- DLORDER, IPADDR, NNSPREF, and TCPNAME start options
- Subarea Mapping Table section

Changed Information

- PARSESS operand under the Application Program Major Node section
- MAXBFRU operand under the Channel-Attachment Major Node section
- Buffer Pool values in the Start Options section

Chapter 1. Introduction

About This Chapter

This chapter gives an overview of VTAM resource definition and presents general information on:
- Major nodes
- Routing functions
- Tables
- Coding conventions, formats, and procedures

Defining a single-domain network to VTAM involves defining the following resources:

- Adjacent control point major nodes
- Application program major nodes
- Channel-attachment major nodes
- External communication adapter major nodes
- Local non-SNA major nodes
- Local SNA major nodes
- LU group major nodes
- Model major nodes
- NCP major nodes
- Switched major nodes
- Path statements [1]
- Transmission group profiles
- Tables

Defining a multiple-domain network to VTAM also involves defining:
- Adjacent SSCP tables [1]
- Cross-domain resource manager major nodes
- Cross-domain resource major nodes [2]
- Network node server lists
- Transport resource list

When you use SNA network interconnection, you might also have to redefine existing major node definitions.

VTAM uses tables for routing, to begin and end sessions, and to define messages and certain operator commands and to suppress messages. You can create, replace, or modify these tables while VTAM is running.

VTAM also provides a filter for session awareness (SAW) data that is passed to communication network management (CNM) application programs, such as NetView. You can create, modify or replace the filter while VTAM is running.

After you define a network, you can use dynamic reconfiguration to add, move, or delete resources attached to an NCP or to add or delete locally attached logical units. In addition, you can dynamically change the values of certain operands.

1. These definition statements, along with dynamic reconfiguration and dynamic path update, are described in “Routing and Dynamic Reconfiguration” on page 481.

2. The cross-domain major node is also used to predefine independent LUs in a boundary-attached subarea network.

© Copyright IBM Corp. 1984, 1999

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on the methods of dynamic reconfiguration.

Major Nodes

Resources in a VTAM network are classified under major and minor nodes. Each major node can contain one or more minor nodes. Brief descriptions of the major nodes follow.

Adjacent Control Point Major Node

An adjacent control point major node specifies the name and attributes of an adjacent APPN® node with which a node can establish type 2.1 connections.

Application Program Major Node

An application program major node contains application programs as minor nodes. There can be one or more application programs in each application program major node.

Channel-Attachment Major Node

A channel-attachment major node contains definition statements for one of several types of channel (link) groups. The possible types of link groups are:

- Channel-to-channel support
- Multipath channel support
- Channel-attached NCP support

Cross-Domain Resource Major Node

A cross-domain resource major node is a logical set of cross-domain resources (CDRSCs). You can define one or more cross-domain resources, each of which is a minor node, in each major node. You can also use the cross-domain resource major node to predefine independent LUs in a boundary-attached subarea network.

Cross-Domain Resource Manager Major Node

A cross-domain resource manager major node is a logical set of cross-domain resource managers (CDRMs). You can define one or more cross-domain resource managers, each of which is a minor node, in each major node.

External Communication Adapter Major Node

An external communication adapter major node defines a connection (port) between VTAM and a token-bus network, a token-ring network, a CSMA/CD 802.3 LAN, or an FDDI LAN attached through the IBM 3172 Nways® interconnect controller. The minor nodes are the line groups, lines, and physical units attached to the port.

Local Non-SNA Major Node

A local non-SNA major node consists of a logical group of channel-attached non-SNA terminals. Each terminal in the group is a minor node. The terminals in a group do not have to all be connected to the same cluster controller.

Local SNA Major Node

A local SNA major node consists of a logical group of channel-attached SNA terminals. The minor nodes are the physical units and their logical units that are contained in the terminals.

LU Group Major Node

An LU group major node defines one or more model LU groups.

Model Major Node

A model major node defines dynamic switched definitions for PUs and LUs.

NCP Major Node

An NCP major node consists of the resources attached to an NCP. The attached resources (lines, physical units, and logical units) are minor nodes.

Switched Major Node

The switched major node contains definitions for potential switched connections to either a subarea node or a peripheral node.

Two types of resources make up a switched connection:

- A switched line within a switched line group

  Define the line group within an NCP major node, a channel-attachment major node, or an external communication adapter major node.
- A remote physical unit

  The physical unit and its logical units are the minor nodes that you define within a switched major node.

TCP/IP Major Node

The VTAM AnyNet host feature permits SNA-to-SNA communication over a Transmission Control Protocol/Internet Protocol (TCP/IP) network. The specific configurations it supports include the following:

- VTAM-to-VTAM communication over a single IP network
- VTAM-to-Communications Manager/2 communication over a single IP network
- VTAM-to-AIX SNA server communication over a single IP network
- VTAM as a multiprotocol transport networking (MPTN) gateway between an SNA network and an IP network

The following must be defined for SNA access to the IP network:

- A TCP/IP major node
- A CDRSC definition for the destination LU in the IP network.

For more information on defining the TCP/IP major node, see *OS/390 eNetwork Communications Server: AnyNet SNA over TCP/IP* .

Transport Resource List Major Node

The transport resource list major node describes the connectivity characteristics of a multipath channel connecting two APPN hosts. The characteristics of each multipath channel are defined by a TRLE definition statement. When an adjacent link station is activated, the PU definition statement identifies which TRLE definition statement VTAM uses to route data over the channel.

Routing Functions and Dynamic Reconfiguration

Brief descriptions of routing functions and dynamic reconfiguration and change follow.

Adjacent SSCP Tables

The adjacent SSCP table contains one or more lists of SSCPs that can be in session with the VTAM SSCP and can be used to reach destination SSCPs in the same or other networks. You can activate one or more adjacent SSCP tables in each VTAM host.

Because of the dynamic adjacent SSCP table function (see the start option “DYNASSCP” on page 587), you do not need to code adjacent SSCP tables to establish cross-domain and cross-network sessions. VTAM dynamically routes session establishment requests to all active adjacent SSCPs until it finds the correct SSCP. However, if you code adjacent SSCP tables, VTAM uses them.

Information on coding the statements that define adjacent SSCP tables is in “Adjacent SSCP Table” on page 501.

Path Definition Statements

Path definition statements are used for two different purposes:

- To define VTAM paths

  PATH definition statements represent the routes VTAM takes to communicate with other subarea nodes. One or more PATH definition statements are filed in the VTAM definition library. The name assigned to a path definition set is used to activate that set of paths. More than one path definition set can be filed and activated.

  Information on coding the definition statements that define VTAM paths is in “VTAM Routes” on page 481.
- For dynamic path update

  Dynamic path update allows the VTAM operator to add NCP path definitions or VTAM path definitions dynamically without regenerating and reloading the NCP. If the explicit route (ER) is not currently operative, NCP path definitions can also be replaced dynamically without regenerating and reloading the NCP. Also, a delete function for inoperative ERs is provided for both VTAM and NCP path definitions. You can also use dynamic path update to change path definition parameters such as virtual route (VR) pacing window size for inoperative routes.

  – For VTAM:

    The dynamic path specification set for VTAM contains VPATH and PATH definition statements.

The VPATH definition statement identifies the VTAM subarea and the network to which the path update specification set applies in a dynamic path update member. The PATH definition statement defines the explicit routes, virtual routes, and virtual route pacing window size for the path.

Information on coding the definition statements that define VTAM dynamic path update is in “Dynamic Path Update (VTAM)” on page 490.

– For NCP:

The dynamic path specification set for an NCP contains NCPPATH and PATH definition statements.

The NCPPATH definition statement identifies the NCP and the network to which the path update specification set applies in a dynamic path update member. The PATH definition statement defines the explicit routes, virtual routes, and virtual route pacing window size for the path.

Information on coding the definition statements that define NCP dynamic path update is in “Dynamic Path Update (NCP)” on page 485.

Dynamic Reconfiguration and Dynamic Change

Dynamic reconfiguration (DR) allows you to change a network configuration without having to deactivate the affected major node and without having to regenerate the NCP. Dynamic change allows you to change certain operands in resource definition statements without having to deactivate the major node. With dynamic reconfiguration and dynamic change, you can:

- Add and delete physical units or logical units
- Move a physical unit and its associated logical units
- Change the SDLC link station address for a physical unit
- Change an LU’s LOCADDR value
- Modify operands

More information on dynamic reconfiguration and the definition statements that are coded for this function is found in “Dynamic Reconfiguration and Dynamic Change” on page 494.

Transmission Group Profile

The transmission group profile definition statement defines a set of characteristics for a transmission group (TG). The TG profile is referenced by an adjacent link station definition on a PU in a switched or model major node or on a PU under a nonswitched line.

The TG profile is not a resource, but describes the characteristics of a link. When an adjacent link station is activated, VTAM locates the TG profile. If the TG profile cannot be found, default values for a TG profile are used for the link characteristics. See “APPN Transmission Group Profile” on page 513 for more details.

Network Node Server List

A network node server list indicates the network nodes that can act as servers for an end node. The network node server list is not required, but using it allows you to control which network node is selected by an end node.

An end node becomes aware of adjacent network nodes that can act as its network node server through normal, spontaneous network flows. By default, an end node

establishes a CP-CP session with the first network node of which it becomes aware, and this network node acts as the end node's server. There is no predictability in this scheme because it depends on timing.

You might want to control which network node is selected by an end node. For example, you might want to exclude a particular network node from network node server responsibilities; acting as a server does involve some overhead, such as originating search requests or executing domain broadcasts. You might also want to isolate particular end nodes from certain network nodes for security reasons.

See "Network Node Server List" on page 529 for more details.

User-Defined Tables and Session Awareness Data Filter

Brief descriptions of the user-defined tables and filter follow.

Associated LU Table

The associated LU table provides the subsystem with the associated printer information needed to dynamically define the SLU. During session initiation, VTAM passes the information to the primary logical unit (PLU) in the LOGON exit.

See "Associated LU Table" on page 661 for information on the macroinstructions that define an associated LU table. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the table is used.

Class of Service

You can group routes based on characteristics such as security, transmission priority, and bandwidth of the routes. The characteristics of a particular group determine a class of service (COS). VTAM provides classes of service for subarea and APPN routing.

See "Subarea Class-of-Service Table" on page 664 for information on the macroinstructions that define a subarea COS table and "APPN Class-of-Service Definitions" on page 668 for information on the statements that define an APPN COS. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of how classes of service are used.

CMIP Services Directory Definition File

The IBM-supplied directory definition file is required to implement Common Management Information Protocol (CMIP) services. The directory definition file for CMIP services controls the associations that CMIP services on a host has with other CMIP services and with itself. An association is a connection between CMIP services on a host and CMIP Services on another node or between CMIP services on the host and itself.

If you have specified the OSIMGMT=YES start option, a directory definition file is required. The IBM-supplied directory definition file in member ACYDDF of the SYS1.SISTCMIP data set allows unlimited associations with other instances of

CMIP services and between CMIP services and itself. If you do not edit the IBM-supplied file, any CMIP application program in any network can have access to CMIP services on this host.

See “CMIP Services Directory Definition File” on page 681 for information on coding the directory definition file. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on the directory definition file.

Interpret Table

When VTAM processes a formatted Initiate or Terminate request (either received directly from a logical unit or formatted by unformatted system services (USS) from a character-coded logon or logoff), it uses an interpret table to determine which application program is to be notified.

The standard logon command definition should suffice for most networks. See “LOGON Command” on page 758 for coding details. You will have to write your own interpret table only under special circumstances, for instance, if the logon sequence you want to use does not follow the syntax for USS commands. IBM does not supply a default table.

See “Interpret Table” on page 690 for information on coding the macroinstructions that define an interpret table. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the table is used.

Logon Mode Table

A logon mode table contains entries specifying session parameters. VTAM contains an IBM-supplied default logon mode table, ISTINCLM. This table contains only a set of generally accepted session parameters for a basic list of IBM device types. If you need session parameters that are not included in the IBM-supplied default table, you can:

- Modify the IBM-supplied default table.
- Create supplementary tables, which you associate with device-type logical units or application programs by using the MODETAB operand in the statements defining them.

See “Logon Mode Table” on page 698 for information on the macroinstructions that define a logon mode table. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the table is used.

Message-Flooding Prevention Table

The message-flooding prevention table lists messages that are candidates for flooding suppression. The purpose is to prevent VTAM from issuing duplicate messages in rapid succession, which could flood the operator console without providing additional information.

For each candidate message, the table contains the criteria that must be met before VTAM suppresses duplicate messages. The suppression criteria include the amount of time between the original and subsequent messages, as well as an indication of

which variable text fields are to be compared and whether messages are suppressed for the hardcopy log. If the message is reissued within the specified interval and the specified variable text fields contain the same information, VTAM suppresses the message.

See “Message-Flooding Prevention Table” on page 714 for information on coding the macroinstructions that define a message-flooding prevention table.

Model Name Table

The model name table provides a model name that is passed to the subsystem in the LOGON exit during session initiation. The subsystem uses the model name while dynamically defining the secondary logical unit (SLU).

See “Model Name Table” on page 720 for information on the macroinstructions that define a model name table. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the table is used.

Session Awareness (SAW) Data Filter

The SAW data filter reduces the amount of session awareness trace data passed over the communication network management (CNM) interface to a CNM application program. See “Session Awareness (SAW) Data Filter” on page 725 for information on the macroinstructions that define a SAW data filter. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the filter is used.

Unformatted System Services (USS) Table

VTAM uses USS tables to define messages and certain operator commands. If you want to change the text or other characteristics of a message or change the syntax or default values for a command, you can do so by creating a supplementary USS table. Only the commands and messages that you want to change need to be defined in the supplementary USS table.

See “Unformatted System Services (USS) Tables” on page 731 for information on the macroinstructions that define a USS table. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the table is used.

Coding VTAM Definitions

This section explains the format of definition statements and macroinstructions and the rules used to code them. For sample definitions to help you implement VTAM functions in your network, refer to *OS/390 eNetwork Communications Server: SNA Resource Definition Samples*.

The VTAMLST definition source validation is available in VTAM as part of the VARY ACT command. It allows you to check the syntax of a definition file without activating the resources in the file. For more information on this function, refer to *OS/390 SecureWay Communications Server: SNA Operation*.

The rules detailed in this section are assembler language rules.

Format of Definition Statements

NCP and VTAM definition statements have the following format:

Name	Definition Statement	Operands
name	Definition statement	Required and optional operands

The *name* field identifies the definition statement or minor node. If you use *name*, it must contain 1–8 characters in the following format:

First character	Alphabetical (A–Z); the national characters @, #, or $; or, for APPL definition statements, the wildcard character ?.
Second to eighth character	Alphabetical (A–Z); numerical (0–9); the national characters @, #, or $; or, for APPL definition statements, the wildcard characters * or ?.

Note: Any *name* fields coded or referenced in VTAM API macroinstructions must be in uppercase.

You can code MVS™ system symbols in the name field of definition statements. See “Using MVS System Symbols in VTAM Definition Statements and Macroinstructions” on page 14 for information about using MVS system symbols in VTAM macroinstructions.

If coded, the *name* field must begin in column 1 of the definition statement and must be followed by one or more blanks. Together, the *name* field and *definition statement* field must fit in columns 1–70 of a line. Although the name might be indicated as optional, you should code it because VTAM uses it in operator messages referring to the resource defined. Also, resources without names are not saved in checkpoint data sets. When a resource is “checkpointed”, information about the resource is saved in a data set and later read by VTAM when it is restarted.

You can use network-qualified names to ensure that resources that are LU-LU session-capable within an SNA network interconnection (SNI) have unique names. Network-qualified names are in the form *netid.resource_name*, where *netid* identifies the name of the network in which the resource resides. Two resources residing in different networks can have the same resource name, but because their network-qualified names have unique network identifiers, no name conflict exists.

The *definition statement* field, which must be preceded and followed by one or more blanks, identifies the definition statement. All definition statements can be in mixed case, even though they will be translated and processed by VTAM in uppercase.

The *operands* field contains operands, which can be coded in any order and must be separated by commas. The operand field ends with one or more blanks placed after the last operand. In most definition statements, operands are used in the operand field. Operands are followed by an equal sign (=) and the operand value. The operand value can be a single value or a list of values. If it is a list of values, the values must be separated by commas and the list must be enclosed in parentheses. If the operand fields are to be continued, the continuation character must be placed in column 72. If the value is specified as *name*, it must follow the preceding rules for *name*.

Note: You can code one or more MVS system symbols in VTAM definition statements in VTAMLST. If you do, the preceding rules for the *name*, *definition statement*, and *operands* fields still apply, even before the symbols are resolved. For example, as shown below, a *name* field containing MVS system symbols must begin in column 1 on the same line as the definition statement to which it applies. The *name* field must be followed by one or more blanks. Together, the *name* field and the *definition statement* field must not go beyond column 70.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
&USER1.&USER2.&USER3.&USER4.&USER5.&USER6.&USER7.&USER8.          APPL X
               AUTH=(ACQ,PASS),ENCR=REQD
```

After MVS substitutes the values for the symbols, the resolved name must be 1–8 characters in length. If the symbols used above are defined in MVS to have the following text:

Symbol	Text
&USER1	A
&USER2	P
&USER3	P
&USER4	L
&USER5	N
&USER6	A
&USER7	M
&USER8	E

the *name* field resolves to APPLNAME after MVS substitutes the values in place of the symbols in VTAMLST.

See “Using MVS System Symbols in VTAM Definition Statements and Macroinstructions” on page 14 for information about using MVS system symbols in VTAM definition statements.

Format of Macroinstructions

VTAM macroinstructions, which define tables and the SAW data filter, have the following format:

Name	Macroinstruction	Operands
name	Macroinstruction	Required and optional operands

The *name* field identifies the macroinstruction. If you use a name, it must contain 1–8 characters in the following format:

First character	Alphabetical (A–Z) or the national characters @, #, or $
Second to eighth character	Alphabetical (A–Z), numerical (0–9), or the national characters @, #, or $.

You can code MVS system symbols in the name field of the macroinstructions that define the following user-definable tables:

- APPN class-of-service (COS) definitions
- APPN-to-subarea class-of-service mapping table
- Associated LU table
- Message-flooding prevention table
- Model name table

- Subarea-to-APPN class-of-service mapping table

See “Using MVS System Symbols in VTAM Definition Statements and Macroinstructions” on page 14 for information about using MVS system symbols in VTAM macroinstructions.

If coded, the name field must begin in column 1 of the macroinstruction, and must be followed by one or more blanks. Together, the *name* field and *macroinstruction* field must fit in columns 1–70 of a line. Although the name might be optional, you should code it, because VTAM uses it in operator messages referring to the resource defined.

The *macroinstruction* field, which must be preceded and followed by one or more blanks, identifies the macroinstruction.

The *operands* field contains operands, which can be coded in any order, and must be separated by commas. The operand field ends with one or more blanks placed after the last operand. In most macroinstructions, operands are used in the operand field. Operands are followed by an equal sign (=) and the operand value. The operand value can be a single value or a list of values. If it is a list of values, the values must be separated by commas and the list must be enclosed in parentheses. If the operand fields are to be continued, the continuation character must be placed in column 72. If the value is specified as *name*, it must follow the rules given above for a name.

Note: You can code one or more MVS system symbols in VTAM macroinstructions in VTAMLST. If you do, the preceding rules for the *name*, *macroinstruction*, and *operands* fields still apply, even before the symbols are resolved. For example, as shown below, a *name* field containing MVS system symbols must begin in column 1 on the same line as the macroinstruction to which it applies. The *name* field must be followed by one or more blanks. Together, the *name* field and the *macroinstruction* field must not go beyond column 70, even when there are no operands to be coded on a macroinstruction, as in the following example.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
&USER1.&USER2.&USER3.&USER4.&USER5.&USER6.&USER7.&USER8.        MDLTAB
```

After MVS substitutes the values for the symbols, the resolved name must be 1–8 characters in length. If the symbols used above are defined in MVS to have the following text:

Symbol	Text
&USER1	M
&USER2	O
&USER3	D
&USER4	L
&USER5	N
&USER6	A
&USER7	M
&USER8	E

the *name* field resolves to MODLNAME after MVS substitutes the values in place of the symbols in VTAMLST.

See “Using MVS System Symbols in VTAM Definition Statements and Macroinstructions” on page 14 for information about using MVS system symbols in VTAM macroinstructions.

Rules for Coding Continued Lines

Code VTAM and NCP definition statements and VTAM macroinstructions —including the name, definition statement or macroinstruction, and operand fields— in columns 1–71 of a line.

Note: The name field together with the definition statement field or macroinstruction field must fit in columns 1–70.

You can continue a definition statement or macroinstruction that exceeds 71 columns on one or more additional lines. When you need to continue on another line, the following rules apply:

1. Code the definition statement or macroinstruction either:
   - Up through column 71, or
   - Through any completed operand, stopping after the comma that separates the operand from those that follow.
2. If the definition statement field or the macroinstruction field ends in column 70, leave a blank in column 71.
3. Enter a nonblank continuation character in column 72. The continuation character is not considered part of the definition statement.
4. Begin coding or continue coding operands in column 16 of the next line. Columns 1–15 must be blank. A continuation line that begins in column 17 or beyond is ignored, with one exception: a quoted string that needs to be continued on the next line can have a blank in column 16 of the next line if a blank in the string coincides with that column.

   A comment line cannot follow a continuation line.
5. If you must continue on another line, proceed with rule 1.
6. Definition statements and macroinstructions can be coded on as many lines as needed.
7. Comments can appear on every line of a continued definition statement or macroinstruction.
8. Columns 73–80 can be used to code identification characters, definition statement sequence characters, or both.

Rules for Coding Comments

You can write comments after the operand field, but they must be separated from the last operand of the operand field by one or more blanks. Comments can be continued to the next line, but they must start in column 17 or beyond. An entire line can be used for a comment by placing an asterisk (*) in the first column of the line. If you need several entire lines for comments, place an asterisk in the first column of each line and leave column 72 blank. A definition statement or macroinstruction that has no operands cannot have comments on the same line as the operation code.

Using a Default

If an operand value has a default, letting the operand's value default or coding the operand's default has the same effect. For example, in the following definition statement for an application program with the name APPL01 the MAXPVT operand was not coded. VTAM will assign the MAXPVT value at 0 because that is the default.

```
APPL01         APPL  APPC=YES,                                    X
               CMPAPPLI=4,                                        X
               CMPAPPLO=1,                                        X
               AUTH=(ACQ,PASS)
```

VTAM will also use the default if you code a null value for the operand. In the following example, the operand was coded without a value.

```
APPL01         APPL  APPC=YES,                                    X
               MAXPVT=
```

Using the Sift-Down Effect in VTAM Definition Statements

The sift-down effect allows you to code an operand on a higher-level node (for example, PU is a higher-level node than LU) so that you do not need to recode it on each lower-level node for which you want the same value. Many of the operands whose values sift to lower-level nodes do not affect the higher-level nodes on which they are coded but can be coded there to take advantage of sifting.

You can override the sifted values at any point in the hierarchy of definition statements. To override the sifted values for a specific lower-level node, use either of the following methods:

- Code the same operand with a different value on the lower-level node. The recoded value applies only to the specific node on which it is coded and all lower-level nodes. All other nodes at the same level will use the sifted value. For example, if you code a sifted value of 2 on an operand of a PU definition statement that has multiple subordinate LU definition statements and then code a value of 4 for the same operand on the third LU statement, the third LU will use the value of 4 and all other LUs subordinate to the PU will use the sifted value of 2.
- Code the same operand with no value, followed by a comma. This causes the default value to replace the sifted value. For example, if you want to automatically logon all but one LU under a PU to application APPL1, you could code LOGAPPL=APPL1 on the PU statement and LOGAPPL=, on the one LU you do not want automatically logged on. The sifted value from the PU definition statement would apply to all LU definition statements except the one on which LOGAPPL=, is coded, whether the other LU definition statements precede or follow it.

The operands to which sifting applies are identified in the definition statement and operand table in each section, in the column labeled "Sift Effect". For information about definition statement sequencing and the sift-down level for each NCP operand, refer to the *NCP, SSP, and EP Resource Definition Reference*.

Note: For resources added through dynamic reconfiguration, sifting takes place within the hierarchy of minor nodes being added. Operand values coded in the original hierarchy do not sift down to those resources dynamically defined.

Restrictions on Names

Specify unique names for all VTAM-related definitions that are active concurrently. For example, the name on a definition statement defining a minor node must not be the major node name. A definition is active when VTAM reads the major node in which it is contained (refer to the *Planning for NetView, NCP, and VTAM* for NCP backup exceptions).

Also, you cannot use any name beginning with “IST” or any of the following names for a major or minor node name:

VTAM
VTAMSEG
VTAMTERM

The names of major node members need to follow the rules for names detailed in “Format of Definition Statements” on page 9 and “Format of Macroinstructions” on page 10, except that they cannot contain the MVS system symbol delimiters, ampersand (&) or a period (.).

Using MVS System Symbols in VTAM Definition Statements and Macroinstructions

You can use MVS system symbols in the names you specify for VTAM definition statements and in the values you specify on the operands on those definition statements in all major nodes and in all definitions for routing and dynamic reconfiguration.

You can also use MVS system symbols in the names you specify for VTAM macroinstructions and in the values you specify on the operands on those macroinstructions for the following user-definable tables:

- Associated LU table
- APPN class-of-service (COS) definitions
- APPN-to-subarea class-of-service mapping table
- Message-flooding prevention table
- Model name table

You can use the following symbols:

&SYSCLONE
Represents a 1–2-character shorthand notation for the name of the system

&SYSNAME
Represents a 1–8-character system name

&SYSPLEX
Represents a 1–8-character sysplex name

&installation-defined system symbol
Represents a 1–8-character identifier. You can define this symbol to be any identifier you want.

The substitution text for these system symbols is defined in MVS and is substituted by MVS in place of the symbols during VTAM startup, major node activation, resource activation, and table activation. Refer to *MVS/ESA SP V5 Initialization and Tuning Reference* for information about how to define the symbols in MVS.

MVS system symbols can be used to allow a single major node, routing or dynamic reconfiguration definition, or user-definable table to be used by multiple VTAMs in a multisystem environment. For information about how to use MVS system symbols in VTAMLST, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* .

Restrictions on Use of Assembler Features

The NCP and VTAM definition statements are coded in standard operating system macroinstruction format, as described in “Coding VTAM Definitions” on page 8, with the following restrictions:

- You cannot use assembler program control instructions (such as ICTL and ISEQ) in groups of definition statements for major nodes.
- You can use assembler listing control statements (such as PRINT, SPACE, and EJECT) in the NCP generation definition statements but not in those for the other types of major nodes.

The following assembler features cannot be used in a group statement for a major node:

- Assembler macroinstructions that generate NCP definition statements
- Names generated by global variables (such as &SYSNDX and &SYSECT)
- Variable substitution at assembly time
- References to assembler attributes (such as length and type)
- Literals
- Quoted strings
- Null keywords. For example, “,,” is treated as a null keyword and is not valid.

You can use comments in groups of definition statements for all types of major nodes and PATH definition statements. However, do not use macroinstructions or operands, such as GENEND, especially in a continued line of comments.

Errors made in the groups of major node definition statements filed in the VTAM definition library result in messages to the system operator’s console during VTAM initialization or VARY ACT command processing.

How to Read Railroad Track Syntax Diagrams

This section describes how to read the syntax diagrams used in this book.

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads (►►) and ends on the right with two arrowheads facing each other (►◄).

```
►►─┤ Syntax Diagram ├──────────────────────────────────────────────►◄
```

- If a diagram is longer than one line, the first line ends with a single arrowhead (►) and the second line begins with a single arrowhead.

```
►►─┤ First Line ├─OPERAND1──OPERAND2──OPERAND3──OPERAND4──OPERAND5──────────►
```

How to Read Railroad Track Syntax

```
►─┤ Second Line ├──────────────────────────────────────────────────►◄
```

- Required operands and values appear on the main path line.

```
►►──REQUIRED_OPERAND──────────────────────────────────────────────►◄
```

  You must code required operands and values.
  If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically in alphanumeric order.

```
►►──┬─REQUIRED_OPERAND_OR_VALUE_1─┬──────────────────────────────►◄
    └─REQUIRED_OPERAND_OR_VALUE_2─┘
```

- Optional operands and values appear below the main path line.

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    └─OPERAND──────┘
```

  You can choose not to code optional operands and values.
  If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.

```
►►──┬──────────────────────────┬──────────────────────────────────►◄
    ├─OPERAND_OR_VALUE_1───────┤
    └─OPERAND_OR_VALUE_2───────┘
```

- An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The comma means that each operand or value must be separated from the next by a comma.

```
      ┌─,──────────────────┐
      ▼                    │
►►────┴─REPEATABLE_OPERAND─┴──────────────────────────────────────►◄
```

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

```
►►──┬──────────────────────────────────────────────────────────┬──►◄
    │   ┌─,──────────────────────────────────────────┐          │
    │   ▼                                            │          │
    └───┬─REPEATABLE_OPERAND_OR_VALUE_1─┬──────────┴──────────┘
        └─REPEATABLE_OPERAND_OR_VALUE_2─┘
```

- A word in all uppercase is an operand or value you must spell exactly as shown. In this example, you must code ***OPERAND***.

  Note: VTAM commands are not case sensitive. You can code them in uppercase or lowercase.

```
►►──OPERAND──────────────────────────────────────────────────────────►◄
```

If an operand or value can be abbreviated, the abbreviation is discussed in the text associated with the syntax diagram.

- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code ***OPERAND=(001,0.001)***.

```
►►──OPERAND=(001,0.001)──────────────────────────────────────────────►◄
```

- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code ***OPERAND=(001 FIXED)***.

```
►►──OPERAND=(001 FIXED)──────────────────────────────────────────────►◄
```

- Default operands and values appear above the main path line. VTAM uses the default if you omit the operand entirely.

```
      ┌─DEFAULT─┐
►►────┤         ├──────────────────────────────────────────────────►◄
      └─OPERAND─┘
```

- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

```
►►──variable─────────────────────────────────────────────────────────►◄
```

- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

```
           (1)
►►──OPERAND──────────────────────────────────────────────────────────►◄
```

Notes:

1 An example of a syntax note.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

```
►►──┤ Reference to Syntax Fragment ├──────────────────────────────────►◄
```

How to Read Railroad Track Syntax

Syntax Fragment:

```
├──1ST_OPERAND,2ND_OPERAND,3RD_OPERAND──────────────────────────────────────┤
```

Filing Definition Statements

To define major nodes and file the definitions in the VTAM definition library or to update your definitions after filing them in the VTAM definition library, do the following:

1. Code the appropriate VTAM or NCP definition statements to define the major node. If you are defining an NCP, code the NCP-only definition statements and operands as described in the NCP resource definition books.

   You must use domain-unique names in all VTAM-related definitions that are active at the same time. For example, the name on a definition statement defining a minor node must not be the major node name. A definition is active when VTAM reads the major node in which it is contained. VTAM reads a major node when it is referenced in the configuration list or when it is activated by a VARY ACT command.
2. Generate the NCP as described in the *NCP, SSP, and EP Generation and Loading Guide*. If you are updating VTAM-only definition statements or operands, update the NCP definitions in the VTAM file; no NCP generation is required. If you are updating NCP-only definition statements or operands, you can use dynamic reconfiguration to update the statements without regenerating the NCP. See “Dynamic Reconfiguration and Dynamic Change” on page 494 for more information.
3. Store the network definition in the proper VTAM library. You can link SYS1.VTAMLST files, but one of the following conditions must be met:
   - The block size must be the same for all files.
   - The file with the largest block size must be the first file in the link.
   - DCB=BLKSIZE=[largest_blocksize_of_ concatenated_files] must be coded on the first file in the link.

   You must run an operating system utility program to put the definitions into the definition library.

Filing Macroinstructions

To create, replace, or modify VTAM’s interpret, logon mode, class-of-service, or USS tables, follow these steps:

1. Code the appropriate VTAM macroinstructions.
2. Assemble the group of macroinstructions.
3. Link-edit the table as a nonexecutable module into a library associated with the VTAMLIB DD statement.

No assembly or link-edit is required to install the associated LU, model name, or message-flooding prevention tables. The groups of macroinstructions are stored in source-language form. To install these tables, follow these steps:

1. Code the appropriate VTAM macroinstructions.
2. Store the group of macroinstructions as a member in SYS1.VTAMLST. The member name is used as the name of the table.

Chapter 2. Major Nodes

Major Nodes

About This Chapter

This chapter contains descriptions of the definition statements, operands, and syntax to define major node resources to the VTAM program.

The major nodes are:

- Adjacent control point
- Application program
- Channel attachment
- Cross-domain resource
- Cross-domain resource manager
- External communication adapter
- Local non-SNA
- Local SNA
- LU group
- Model
- Network Control Program (NCP)
- Switched
- Transport resource list

Adjacent Control Point Major Node

To define an adjacent control point (ADJCP) major node in an APPN network, code a VBUILD definition statement for the major node and an ADJCP definition statement for each adjacent control point (CP) minor node in the major node. There can be one or more minor nodes in the major node.

Note: If the DYNADJCP start option is defaulted or specified as YES, VTAM dynamically creates ADJCP definition statements in the dynamic adjacent control point major node (ISTADJCP) for each adjacent control point.

If the DYNADJCP start option is defaulted or specified as YES, and you code an ADJCP major node, adjacent control points not specified in the ADJCP major node are still dynamically defined in the ISTADJCP major node.

Table 1. Adjacent Control Point Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=ADJCP	24	R			APPN only
name	ADJCP	DYNLU	21			DYNLU start option	APPN only
		NATIVE	23				BN=YES; APPN only
		NETID	23				APPN only

© Copyright IBM Corp. 1984, 1999

Table 1. Adjacent Control Point Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		NN	24				APPN only; VN
		VN	24			NO	APPN only; DYNLU=NO; NN

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each adjacent control point major node.

ADJCP Definition Statement

Code one ADJCP definition statement for each adjacent control point (CP) in the major node.

NODETYPE Start Option

Note that the NODETYPE start option is required for this major node.

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start options determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

Full Syntax

The full syntax for the adjacent control point major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=ADJCP──────────────────────────────────►◄
    └─name─────┘
```

ADJCP

```
                    ┌─DYNLU=value_of_dynlu_start_option──────┐
►►─name──ADJCP──┬───┼──────────────────────────────────────┼──────────►
                │   │                ┌─value_of_dynlu_start_option─┐
                │   └─DYNLU=─────────┼─────────────────────────────┼──┘
                │                    ├─NO──────────────────────────┤
                │                    └─YES─────────────────────────┘
                │   ┌─DYNLU=YES─┐  (1)
                └───┼───────────┼──────────────────────────────────┘
                    └─DYNLU=YES─┘
```

```
►──┬─NATIVE=─┬─NO──┬──┬─NETID=network_id─┬──┬─NN=─┬─NO──┬──►
   │         │     │  │                  │  │     │     │
   │         └─YES─┘  │                  │  │     └─YES─┘
   │                  │                  │  │
   ►──┬─VN=NO──────┬──────────────────────────────────────►◄
      │            │
      └─VN=─┬─NO──┬──┘
            │     │
            └─YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

Operand Descriptions

The following section describes the operands you can code for the adjacent control point major node.

DYNLU

```
      ┌─DYNLU=value_of_dynlu_start_option──────┐
►►────┼──────────────────────────────────────────┼──────────────►◄
      │          ┌─value_of_dynlu_start_option─┐│
      └─DYNLU=───┼─NO──────────────────────────┼┘
                 └─YES─────────────────────────┘
      ┌─DYNLU=YES──┐  (1)
      │            │
      └─DYNLU=YES──┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statement:	*ADJCP*
dependencies:	*APPN only*

Specifies whether cross-domain resources using adjacent link stations attached to this adjacent control point can be defined dynamically. The value coded here overrides the value coded on the DYNLU start option and also overrides the values coded on definition statements for resources attached to this adjacent CP.

DYNLU=NO
Specifies that dynamic definition of CDRSCs is not allowed for this adjacent CP. You must predefine resources that use adjacent link stations attached to this adjacent CP. If you do not define the resources, the session request will fail.

DYNLU=YES
Specifies that dynamic definition of CDRSCs is allowed for this adjacent CP. You are not required to predefine resources that use adjacent link stations attached to this adjacent CP.

For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

Note: To ensure predictable results, you should code DYNLU on the ADJCP definition statement. If you do not, the DYNLU value coded on the first activated adjacent link station (ALS) is assigned to this adjacent CP and to all resources attached to the ALS.

If DYNLU is not coded on the first activated adjacent link station, the value of the DYNLU start option, which defaults to NO, is assigned to this adjacent CP and to all resources attached to this adjacent CP. The DYNLU value taken by an adjacent CP and all resources attached to it depends on where you code the DYNLU operand. Table 2 shows the DYNLU coding combinations.

Table 2. DYNLU Combinations

DYNLU Coded On:						**DYNLU value assigned to adjacent CP and attached resources:**
ADJCP	**First PU Activated**	**LINE**	**GROUP**	**Other PUs**	**Start Option**	
DYNLU Coded	No effect	No effect	No effect	No effect	No effect	**Value coded on ADJCP**
DYNLU Not Coded	**DYNLU Coded**	No effect	No effect	No effect	No effect	**Value coded on first PU**
DYNLU Not Coded	DYNLU Not Coded	**DYNLU Coded**	No effect	No effect	No effect	**Value coded on LINE (sifts to PU)**
DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	**DYNLU Coded**	No effect	No effect	**Value coded on GROUP (sifts to PU)**
DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	**DYNLU Coded**	No effect	**Value coded on DYNLU start option, or default (DYNLU=NO)**
DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	**DYNLU Coded**	**Value coded on DYNLU start option**
DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	DYNLU Not Coded	**Default DYNLU start option (DYNLU=NO)**

name

```
►►──┬──────┬──────────────────────────────►◄   ►►──name──────────────────────────────►◄
    └name──┘
```

statements: VBUILD *statements:* ADJCP

On VBUILD: Specifies the optional name of this major node.

If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On ADJCP: Specifies the required name of the adjacent CP to which connections are to be established.

NATIVE

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └NATIVE=──┬──NO──┬──────────┘
              └──YES─┘
```

statements:	*ADJCP*
dependencies:	*BN=YES; APPN only*

Specifies whether the adjacent node is expected to be in the native subnetwork. If NATIVE is not specified on the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

No subnetwork negotiations will occur if NATIVE has been specified, in this case the value between the two nodes must match for connection to occur.

If you code NATIVE on both the ADJCP and PU definition statements, the values must match.

For transmission groups (TGs) between nodes, the value specified or negotiated for NATIVE must match. If a TG is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the transmission group attempting activation is not allowed to become active. If the activation is for a virtual-route-based transmission group then the SSCP-SSCP session will also fail.

NETID

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └NETID=network_id──────────┘
```

statement:	*ADJCP*
dependencies:	*APPN only*

The network identifier of the adjacent CP.

If NETID is specified, then the network identifier is validated when a connection is established to the adjacent node. If the NETIDs do not match, the session setup is based on the value of the DYNADJCP start option. If DYNADJCP=YES, the session is established dynamically; if DYNADJCP=NO, the session is not established.

If NETID is not specified, the adjacent node's network identifier is acknowledged at connection establishment. When an ADJCP is defined without a NETID, any NETID with that ADJCP name is considered to be predefined, so the session is always established dynamically, even if DYNADJCP=NO is specified in the start options.

NN

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └─NN=──┬─NO──┬─────┘
           └─YES─┘
```

statement:	*ADJCP*
dependencies:	*VN; APPN only*

Notes:

1. NN and VN on the ADJCP definition statement on the ADJCP major node are mutually exclusive.

Specifies whether the adjacent node is expected to be a network node.

If you specify a value for NN, it is validated when a connection is attempted to the adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If you do not specify a value for NN, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

Note: NN and VN are mutually exclusive.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

TYPE

```
►►──TYPE=ADJCP──────────────────────────────────────────────────────────►◄
```

statement:	*VBUILD*
dependencies:	*APPN only*

Specifies that the VBUILD statement defines an adjacent control point major node to VTAM.

VN

```
►►──┬─VN=NO──────────────┬──────────────────────────────────────────────►◄
    │                    │
    └─VN=──┬─NO──┬───────┘
           └─YES─┘
```

statements:	*ADJCP*

dependencies: *NN; DYNLU=NO; APPN only*

Specifies that a virtual node is to be predefined.

VN=NO
Specifies that no virtual node is to be predefined.

VN=YES
Specifies that a virtual node is to be predefined.

Application Program Major Node

To define an application program major node, code a VBUILD definition statement for the major node, an optional GROUP definition statement to group application program minor nodes, and an APPL definition statement for each application program minor node in the major node. There might be one or more application programs in each application program major node.

Registering Application Programs

An APPN node by default provides some level of resource registration within the network node or end node, depending on its resource type. However, the level of registration can be specified on local resource definitions with the REGISTER operand.

Because VTAM application programs are likely to generate more LU-LU session traffic than other resources, and therefore more APPN locate searches, application programs should be registered on their network node server and optionally on the central directory server to reduce traffic. However, in certain cases, an application program should not be dynamically defined to the APPN network. Such application programs include:

- Programs without network-unique names, such as those that have been excluded from participating in cross-domain session setup
- Programs that participate only in LU-LU sessions within the local VTAM domain
- Programs that can participate in cross-domain LU-LU sessions but will never be the target of a cross-domain search request.

An example of an application program that should not be defined dynamically is TSO/VTAM. TSO usually runs on a host as several applications programs. One main task handles cross-domain LU-LU session setup, and several subtasks control LU-LU sessions after session setup so that TSO services can be accessed by users. Refer to the “Resource Registration in an APPN Network” in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how TSO/VTAM applications are registered.

Non-LU 6.2 Application Programs

Table 3. Application Program Major Node (Non-LU 6.2) Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=APPL	62	R			
name	GROUP	See the APPL definition statement for operands.					
name	APPL	ACBNAME	33			*name* value	

Table 3. Application Program Major Node (Non-LU 6.2) Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		APPC	34		G	NO	
		ASLENT	34		G		ASLTAB
		ASLTAB	35		G		
		ASRCVLM	35		G		
		AUTH	36		G	(NOACQ, NOASDP, NOCNM, NOPASS, NOPO, NOTSO, VPACE)	
		CERTIFY	38		G	NO	ENCR
		CMPAPPLI	39		G	0	
		CMPAPPLO	39		G	0	
		DLOGMOD	40		G		MODETAB
		EAS	43		G	509	
		ENCR	44		G	OPT	
		ENCRTYPE	45		G	DES	
		FASTPASS	46		G	YES	TSO only
		HAVAIL	46		G	NO	
		LOSTERM	48		G	NORMAL	
		LUAPFX	48				
		MAC	49		G	NONE	
		MACLNTH	49		G	4	
		MACTYPE	50		G		
		MAXPVT	50		G	0	
		MDLENT	51		G		MDLTAB
		MDLTAB	51		G		
		MODETAB	52		G	ISTINCLM	
		MODSRCH	52		G	NEVER	model only
		PARSESS	55		G	NO	APPC=NO
		PERSIST	55			SINGLE	
		POAQLIM	56		G		AUTH
		PRTCT	56		G		
		REGISTER	57		G	CDSERVR	APPN only
		SESSLIM	59		G	NO	
		SONSCIP	59		G	NO	
		SPAN	60		G		NetView
		SRBEXIT	60		G	NO	
		SSCPFM	60		G	USSPOI	AUTH
		USSTAB	62		G	ISTINCNO	AUTH
		VCNS	62		G	NO	
		VPACING	64		G	7	
		VTAMFRR	64		G	NO	

Table 3. Application Program Major Node (Non-LU 6.2) Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

LU 6.2 Application Programs

Table 4. Application Program Major Node (LU 6.2) Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=APPL	62	R			
name	GROUP	See the APPL definition statement for operands.					
name	APPL	ACBNAME	33			*name* value	
		APPC	34		G	NO	APPC=YES
		ASLENT	34		G		ASLTAB
		ASLTAB	35		G		
		ATNLOSS	35		G	LAST	APPC=YES
		AUTH	36		G	(NOACQ, NOASDP, NOCNM, NOPASS, NOPO, NOTSO, VPACE)	
		AUTOSES	38		G	0	APPC=YES
		CMPAPPLI	39		G	0	
		CMPAPPLO	39		G	0	
		DDRAINL	40		G	NALLOW	APPC=YES
		DLOGMOD	40		G		MODETAB
		DMINWNL	41		G	1	APPC=YES DMINWNR DSESLIM
		DMINWNR	41		G	1	APPC=YES; DMINWNL; DSESLIM
		DRESPL	42		G	NALLOW	APPC=YES
		DSESLIM	42		G	2	APPC=YES; DMINWNL; DMINWNR
		DSPLYWLD	43		G	YES	AUTH
		EAS	43		G	509	
		ENCR	44		G	OPT	
		ENCRTYPE	45		G	DES	
		FASTPASS	46		G	YES	TSO only
		HAVAIL	46		G	NO	
		LIMQSINT	47		G		APPC=YES
		LMDENT	47		G	19	APPC=YES

Table 4. Application Program Major Node (LU 6.2) Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		LOSTERM	48		G	NORMAL	
		MAXPVT	50		G	0	
		MAC	49		G	NONE	
		MACLNTH	49		G	4	
		MACTYPE	50		G	4	
		MDLENT	51		G		MDLTAB
		MDLTAB	51		G		
		MODETAB	52		G	ISTINCLM	
		MODSRCH	52		G	NEVER	model only
		OPERCNOS	54		G	NALLOW	APPC=YES
		PARSESS	55		G	YES	APPC=YES
		PERSIST	55			SINGLE	
		POAQLIM	56		G		AUTH
		PRTCT	56		G		
		REGISTER	57		G	CDSERVR	APPN only
		SECACPT	57		G	NONE	APPC=YES
		SECLVL	58		G	ADAPT	APPC=YES
		SESSLIM	59		G	NO	
		SONSCIP	59		G	NO	
		SPAN	60		G		NetView
		SRBEXIT	60		G	NO	
		SSCPFM	60		G	USSPOI	AUTH
		SYNCLVL	61		G	CONFIRM	APPC=YES
		USSTAB	62		G	ISTINCNO	AUTH
		VCNS	62		G	NO	
		VERIFY	63		G	NONE	APPC=YES
		VPACING	64		G	7	
		VTAMFRR	64		G	NO	

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each application program major node.

GROUP Definition Statement

Code a GROUP definition statement to group application program minor nodes. All operands that are valid on the APPL definition statement can be coded instead on the GROUP statement for sifting, except for ACBNAME, which must be coded on the APPL definition statement.

Note: If APPC is coded on the GROUP definition statement, its value cannot be overridden by the APPL definition statement.

APPL Definition Statement

Code one APPL definition statement for each application program that is in the VTAM domain and include it in an application program major node.

If you use TSO/VTAM, the MVS logon manager, or the NetView alias translation facility for alias name translation, there are specific coding requirements for the APPL definition statement. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for specific information on these requirements.

Full Syntax

The full syntax for the application program major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=APPL──────────────────────────────────────►◄
    └name──┘
```

GROUP

```
►►──name──GROUP──────────────────────────────────────────────────────►◄
```

APPL

```
                   ┌ACBNAME=value_of_name_operand┐   ┌APPC=NO──────┐
►►──name──APPL─────┤                             ├───┤             ├──►
                   └ACBNAME=acb_name─────────────┘   └APPC=──┬NO──┬──┘
                                                             └YES─┘

►──┬──────────────────────────────────────────┬──────────────────────►
   └ASLENT=associated_lu_table_entry_name─────┘

                                   ┌ASRCVLM=1,000,000──────────────┐
►──┬──────────────────────────────┤                               ├──►
   └ASLTAB=associated_lu_table_name┘ └ASRCVLM=──┬amt_of_address_space┬┘
                                                └1,000,000───────────┘
```

Application Program Major Node

```
►─┬─ATNLOSS=LAST──────────┬──────────────────────────────────────►
  │                       │
  └─ATNLOSS=─┬─LAST─┬─────┘
             └─ALL──┘

►─┬─AUTH=(NOACQ,NOASDP,NOCNM,NOPASS,NOPO,NOTSO,VPACE)─┬──────────►
  │                                                   │
  └─AUTH=(──┬─NOACQ─┬──────────────────────────────────┬──)──────┘
            │       │                                  │
            │       └─ACQ───┘                          │
            │─NOASDP─┬──────────────────────────────────│
            │       │                                  │
            │       └─ASDP──┘                          │
            │─NOCNM──┬──────────────────────────────────│
            │       │                                  │
            │       └─CNM───┘                          │
            │─NOPASS─┬──────────────────────────────────│
            │       │                                  │
            │       └─PASS──┘                          │
            │─NOPO───┬──────────────────────────────────│
            │       │                                  │
            │       ├─PPO───┤                          │
            │       └─SPO───┘                          │
            │─NOTSO──┬──────────────────────────────────│
            │       │                                  │
            │       └─TSO───┘                          │
            │─VPACE──┬──────────────────────────────────│
            │       │                                  │
            │       └─NVPACE┘                          │

►─┬─AUTOSES=0──────────────────┬─┬─CERTIFY=NO──────────┬──────────►
  │                            │ │                     │
  └─AUTOSES=number_of_sessions─┘ └─CERTIFY=─┬─NO─┬────┘
                                            └─YES┘

►─┬─CMPAPPLI=0──────────────────────────┬──────────────────────────►
  │                                     │
  └─CMPAPPLI=input_data_compression_level┘

►─┬─CMPAPPLO=0──────────────────────────┬─┬─DDRAINL=NALLOW──────┬──►
  │                                     │ │                     │
  └─CMPAPPLO=output_data_compression_level┘ └─DDRAINL=─┬─NALLOW─┬──┘
                                                      └─ALLOW──┘

►─┬─DLOGMOD=default_logon_mode_table_entry_name─┬──────────────────►
  │                                             │
  └─────────────────────────────────────────────┘
```

```
►─┬─DMINWNL=1──────────────────────┬─┬─DMINWNR=1──────────────────────┬──────────►
  └─DMINWNL=number_of_sessions─────┘ └─DMINWNR=number_of_sessions─────┘

►─┬─DRESPL=NALLOW──────────┬──────────────────────────────────────────────────►
  │                        │
  └─DRESPL=─┬─NALLOW─┬─────┘
            └─ALLOW──┘

►─┬─DSESLIM=2──────────────────────┬─┬─DSPLYWLD=YES──────────┬──────────────────►
  └─DSESLIM=number_of_sessions─────┘ │                       │
                                     └─DSPLYWLD=─┬─YES─┬─────┘
                                                 └─NO──┘

►─┬─EAS=509──────────────────────┬─┬─ENCR=OPT──────────┬─┬─ENCRTYPE=DES────┬──────►
  └─EAS=number_of_sessions───────┘ │                   │ └─ENCRTYPE=TDES24──┘
                                   └─ENCR=─┬─OPT──┬────┘
                                           ├─COND─┤
                                           ├─NONE─┤
                                           ├─REQD─┤
                                           └─SEL──┘

►─┬─FASTPASS=YES──────────┬─┬─HAVAIL=NO──────────┬─┬──────────────────────────►
  │                       │ │                    │ └─LIMQSINT=timeout_value─────┘
  └─FASTPASS=─┬─YES─┬─────┘ └─HAVAIL=─┬─NO──┬────┘
              └─NO──┘                └─YES─┘

►─┬─LMDENT=19──────────────────────┬─┬─LOSTERM=NORMAL──────────┬─┬──────────────►
  └─LMDENT=number_of_entries───────┘ │                         │ └─LUAPFX=string──┘
                                     └─LOSTERM=─┬─NORMAL─┬─────┘
                                                ├─IMMED──┤
                                                └─SECOND─┘

►─┬─MAC=NONE──────────────┬─┬─MACLNTH=4──────────┬─┬─MACTYPE=CRC──────────┬──────►
  │                       │ │                    │ │                      │
  └─MAC=─┬─NONE─┬─────────┘ └─MACLNTH=─┬─4──┬─────┘ └─MACTYPE=─┬─CRC──┬─────┘
         ├─COND─┤                     ├─2──┤                   └─DES──┘
         └─REQD─┘                     ├─6──┤
                                      └─8──┘

►─┬─MAXPVT=0──────────────────────┬──────────────────────────────────────────────►
  └─MAXPVT=number_of_bytes─────────┘
```

Application Program Major Node

```
►──┬──────────────
```

```
►─┬─VERIFY=NONE──────────┬─┬─VPACING=7──────────────────┬──────────────►
  │                      │ │                            │
  └─VERIFY=─┬─NONE───────┘ └─VPACING=number_of_requests──┘
            ├─OPTIONAL─┤
            └─REQUIRED─┘

►─┬─VTAMFRR=NO──────┬──────────────────────────────────────────────────►◄
  │                 │
  └─VTAMFRR=─┬─NO──┬┘
             └─YES─┘
```

Notes:

1 PARSESS defaults to NO when APPC=NO.

2 PARSESS defaults to YES when APPC=YES.

Operand Descriptions

The following section describes the operands you can code for the application program major node.

ACBNAME

```
►►─┬─ACBNAME=value_of_name_operand─┬──────────────────────────────────►◄
   │                               │
   └─ACBNAME=acb_name──────────────┘
```

statements:	*APPL*
dependencies:	*name*

Specifies the minor node name assigned to this application program. This name must be unique within the domain in which the application program resides.

This operand is needed if application programs with the same APPLID are started in multiple hosts and the application programs are to set up a cross-domain session. If this name is not identical to the network name, it cannot be network qualified and used in any VTAM command or macroinstruction. The ACBNAME can only be used by an LU (including an application program LU) in a command issued in the same domain where the ACBNAME is defined.

If you do not code this operand, the network-unique name (the name of the APPL definition statement) is used as the ACBNAME.

For application programs used in extended recovery facility (XRF) sessions, the network name and the ACBNAME must be the same. Therefore, do not code the ACBNAME operand; use the default, which is the network name.

acb_name must follow the rules for naming application programs as described in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and can include the following wildcard characters:

Asterisk (*)
Represents 0 or more unspecified characters. An asterisk can be used as the second to eighth character.

Question mark (?)
Represents a single unspecified character. A question mark can be used as the first to eighth character.

For information about how to use wildcard characters to create model application program definitions, refer to “Model Application Program Definitions” in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

APPC

```
►►─┬─APPC=NO──────────┬──────────────────────────────────────────►◄
   │                  │
   └─APPC=─┬─NO──┬────┘
           └─YES─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies whether the application program can issue APPCCMD macroinstructions. Refer to the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference* for a description of this macroinstruction.

Note: APPC, if specified as NO (or defaulted), on the GROUP definition statement, can be overridden on the APPL definition statement with APPC=YES. However, if APPC=YES on the GROUP definition statement, APPC=NO is ignored on the APPL definition statement and all application programs in the group will be APPC-capable.

APPC=NO
Specifies that the application program cannot issue APPCCMD macroinstructions.

APPC=YES
Specifies that the application program can issue APPCCMD macroinstructions.

ASLENT

```
►►─┬──────────────────────────────────────┬──────────────────────►◄
   └─ASLENT=associated_lu_table_entry_name─┘
```

statements: *GROUP, APPL*
dependencies: *ASLTAB*

Specifies the name of the associated LU table entry to be used for this application program.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─ASLTAB=associated_lu_table_name──┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies the name of the associated LU table to be used for this application program. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. Refer to “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about this command.

ASRCVLM

```
      ┌─ASRCVLM=1,000,000──────────────┐
►►────┤                                ├──────────────────────────────────►◄
      └─ASRCVLM=──┬─amt_of_address_space─┬─┘
                  └─1,000,000────────────┘
```

statements: *GROUP, APPL*
dependencies: *none*
range: *0–99,999,999 bytes*

Specifies an amount of address space storage used by VTAM to calculate constrain and critical threshold values for the address space. High usage periods of address space storage should be considered when selecting this value. If you specify 0, the amount of data space storage used is 1,000,000 bytes.

Note: ASRCVLM is applicable only to the application receiving the data and not to the sender. VTAM will queue inbound data to the application’s dataspace if the application does not have a RECEIVE RPL available to field the data. The ASRCVLM parameter is used by the receiving VTAM to adjust the window size in the pacing responses returned to the data sender. Eventually the ASRCVLM limits may be hit and VTAM will respond by lowering the session level.

ATNLOSS

```
►►─┬─ATNLOSS=LAST──────────┬──────────────────────────────────────►◄
   │                       │
   └─ATNLOSS=─┬─LAST─┬──────┘
              └─ALL──┘
```

statements: *GROUP, APPL*
dependencies: *APPC=YES*

Specifies when the ATTN exit should be scheduled.

ATNLOSS=ALL
Specifies that the ATTN exit is to be scheduled for all session deactivations.

ATNLOSS=LAST
Specifies that the ATTN exit is to be scheduled when the last session between an (LU-mode) pair is deactivated.

Code this operand for any LU 6.2 application.

AUTH

```
►►─┬─AUTH=(NOACQ,NOASDP,NOCNM,NOPASS,NOPO,NOTSO,VPACE)─┬──────────►◄
   │                                                   │
   │              ┌─NOACQ─┐                            │
   └─AUTH=(───────┼───────┼──────────────────────)─────┘
                  └─ACQ───┘
                  ┌─NOASDP─┐
                  ├────────┼──────────────────────────
                  └─ASDP───┘
                  ┌─NOCNM─┐
                  ├───────┼──────────────────────────
                  └─CNM───┘
                  ┌─NOPASS─┐
                  ├────────┼──────────────────────────
                  └─PASS───┘
                  ┌─NOPO───┐
                  ├────────┼──────────────────────────
                  ├─PPO────┤
                  └─SPO────┘
                  ┌─NOTSO─┐
                  ├───────┼──────────────────────────
                  └─TSO───┘
                  ┌─VPACE──┐
                  ├────────┼──────────────────────────
                  └─NVPACE─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Determines whether this application program has the authority to use certain VTAM functions. You can code these operands in any order. The possible operands are listed below:

AUTH=ACQ

AUTH=NOACQ

Determines whether this application program can use either the OPNDST macroinstruction with the ACQUIRE option or the SIMLOGON macroinstruction. (These macroinstructions enable the application program to initiate a session with a particular logical unit.)

If you code APPC=YES, the default is ACQ. If you code APPC=YES and AUTH=NOACQ, VTAM supplies an override of ACQ and issues a warning message.

AUTH=ASDP

AUTH=NOASDP

Specifies whether an application can supply dial digits (telephone number) and other parameters for a dial-out switched connection during session initiation.

AUTH=CNM

AUTH=NOCNM

Determines whether this application program can use the communication network management (CNM) interface. This interface allows an application program to send and receive formatted request units from certain physical units in the application program's domain. For more information on the CNM interface, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

You must code AUTH=CNM for all application programs for which entries exist in the IBM-supplied CNM routing table or for any user-written applications using the CNM interface. For a complete list of IBM-supplied CNM applications, refer to *OS/390 SecureWay Communications Server: SNA Customization*.

AUTH=PASS

AUTH=NOPASS

Determines whether this application program can use the CLSDST macroinstruction with the PASS option to pass session-establishment requests to other application programs.

For more information on the CLSDST macroinstruction, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

AUTH=PPO

AUTH=SPO

AUTH=NOPO

Specifies whether the application program can issue VTAM operator commands and receive responses and unsolicited messages. *OS/390 eNetwork Communications Server: SNA Programming* describes the program operator application.

If you code either the primary program operator (PPO) or the secondary program operator (SPO), the application program is authorized to issue SENDCMD and RCVCMD macroinstructions.

If you code PPO, the application program receives all unsolicited messages, that is, all informational and error messages that are not replies to operator commands. If no PPO-designated application program is active when these messages occur, they are directed to the system console.

AUTH=TSO

AUTH=NOTSO
Specifies whether the application program is a TSO/VTAM time-sharing program. If it is a TSO/VTAM time-sharing program, code TSO. If not, code NOTSO or take the default.

If you code AUTH=TSO, you cannot use wildcard values in the name field of this APPL definition statement.

AUTH=VPACE

AUTH=NVPACE
Determines whether this application program is subject to the VPACING specifications of SLUs with which the program is in session. Coding NVPACE is the same as coding VPACING=0 in the LU definition statements for all of the SLUs with which the application program is in session. NVPACE is ignored for same-domain local SNA logical units.

Note: It is recommended that you code AUTH=NVPACE for normal interactive TSO/VTAM sessions. However, if you have application-to-application TSO/VTAM sessions, it is recommended that you pace them by coding VPACING=20.

For more information on pacing, refer to the discussion in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

AUTOSES

```
►►─┬─AUTOSES=0──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─AUTOSES=number_of_sessions─┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*
range:	*0–32767*

Tells VTAM the number of contention-winner sessions to start automatically, rather than by allocation requests from the application program. VTAM establishes the specified number of sessions when the first change number of sessions (CNOS) request is issued for a mode-name group of a given partner LU.

Refer to *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide* for more information about the role of the AUTOSES operand in session activation.

CERTIFY

```
►►─┬─CERTIFY=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   │             ┌─NO──┐ │
   └─CERTIFY=────┴─YES─┘─┘
```

statements:	*GROUP, APPL*

dependencies: *CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner's LU support also supports the CERTIFY function.*

Specifies whether VTAM on behalf of this application program will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CMPAPPLI

```
►►─┬─CMPAPPLI=0──────────────────────────────┬──────────────────────────────────►◄
   │                                         │
   └─CMPAPPLI=input_data_compression_level─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies the maximum compression levels for the application's input data, which is the data the primary logical unit (PLU) receives. The operand allows compression-level limits for a specific application to be set lower than the overall limit specified by the CMPVTAM start option.

Values of *input_data_compression_level* follow:
0 No compression
1 Run-length encoding (RLE)
2 Small adaptive compression table
3 Medium adaptive compression table
4 Large adaptive compression table

The limit specified by the operand is enforced only during sessions for which:
- The application acts as the PLU.
- The limit is lower than that specified by the CMPVTAM start option.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about compression levels.

CMPAPPLO

```
►►─┬─CMPAPPLO=0──────────────────────────────┬──────────────────────────────────►◄
   │                                         │
   └─CMPAPPLO=output_data_compression_level─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies the maximum compression levels for the application's output data, which is the data the primary logical unit (PLU) sends. The operand allows

compression-level limits for a specific application to be set lower than the overall limit specified by the CMPVTAM start option.

Values of *output_data_compression_level* follow:

0 No compression
1 Run-length encoding (RLE)
2 Small adaptive compression table
3 Medium adaptive compression table
4 Large adaptive compression table

The limit specified by the operand is enforced only during sessions for which the application acts as the PLU, and the limit is lower than that specified by the CMPVTAM start option.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about compression levels.

DDRAINL

```
►►─┬─DDRAINL=NALLOW──────────┬──────────────────────────────────────►◄
   │                         │
   └─DDRAINL=─┬─NALLOW─┬─────┘
              └─ALLOW──┘
```

statements: *GROUP, APPL*
dependencies: *APPC=YES*

Specifies whether the application program can drain waiting allocation requests when the application is the target of a CNOS request that resets the session limit to zero.

DDRAINL=ALLOW
Allows the application program to drain its allocation requests. The drain-target value in the received CNOS request is not negotiated.

DDRAINL=NALLOW
Does not allow the application program to drain its allocation requests. The drain-target value in the received CNOS request is negotiated to no draining.

DLOGMOD

```
►►─┬──────────────────────────────────────────────────────────────────►◄
   └─DLOGMOD=default_logon_mode_table_entry_name──────────────────────┘
```

statements: *GROUP, APPL*
dependencies: *MODETAB*

Specifies the name of the logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM.

A logon mode entry determines which entry in the applicable logon mode table is to provide a set of session parameters for the application program if the application program is a secondary logical unit (SLU). The name specified on the DLOGMOD operand must be the name of an entry in a logon mode table.

For more information, see “Logon Mode Table” on page 698.

DMINWNL

```
►►─┬─DMINWNL=1──────────────────────┬──────────────────────────────────────►◄
   │                                │
   └─DMINWNL=number_of_sessions─────┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES, DMINWNR, DSESLIM*
range:	*0–32767*

Defines the minimum number of parallel sessions for a mode name group for a given remote logical unit of which this application program (the local logical unit) is guaranteed to be the contention winner and the remote logical unit is guaranteed to be the contention loser. A mode name group is a group of sessions for a given mode name.

The DMINWNL value must always be less than or equal to the DSESLIM value minus the DMINWNR value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference* for information on how VTAM negotiates the actual number of contention-winner sessions for this application (local logical unit).

DMINWNR

```
►►─┬─DMINWNR=1──────────────────────┬──────────────────────────────────────►◄
   │                                │
   └─DMINWNR=number_of_sessions─────┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES, DMINWNL, DSESLIM*
range:	*0–32767*

Defines the minimum number of parallel sessions for a mode name group for a given remote logical unit of which the remote logical unit is guaranteed to be the contention winner and this application program (local logical unit) is guaranteed to be the contention loser.

The DMINWNR value must always be less than or equal to the DSESLIM value minus the DMINWNL value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference* for information on how VTAM negotiates the actual number of contention-winner sessions for the remote logical unit.

DRESPL

```
►►─┬─DRESPL=NALLOW──────────┬──────────────────────────────────────────►◄
   │                        │
   └─DRESPL=─┬─NALLOW─┬─────┘
             └─ALLOW──┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*

Specifies whether the local application program (local logical unit) will deactivate sessions on receipt of a CNOS request that specifies the application program as the responsible logical unit.

Refer to the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide* for an explanation of draining.

DRESPL=ALLOW
Allows the application program to deactivate sessions. The value indicating the responsible LU in the reserved CNOS is not negotiated.

DRESPL=NALLOW
Does not allow the application program to deactivate sessions. Instead, VTAM assigns the responsibility for deactivating sessions to the remote logical unit that sent the CNOS request.

DSESLIM

```
►►─┬─DSESLIM=2──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─DSESLIM=number_of_sessions─┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES, DMINWNL, DMINWNR*
range:	*0–32767*

Defines the maximum number of sessions to be allowed between the local application logical unit and a remote logical unit on a given mode name group.

The DSESLIM value is always greater than or equal to the combined values for DMINWNL and DMINWNR.

If you code DSESLIM=0 and the application program is the target of a CNOS request to start the session limit for a new mode, VTAM responds to the CNOS request with an indication that the requested mode is not available.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference* for information on how VTAM determines the negotiated session limit.

DSPLYWLD

```
►►─┬─DSPLYWLD=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─DSPLYWLD=─┬─YES─┬─────┘
               └─NO──┘
```

statements: *GROUP, APPL*
dependencies: *AUTH=PPO or AUTH=SPO*

Indicates whether the program operator application can issue DISPLAY commands containing wildcards.

DSPLYWLD=NO
Specifies that wildcards are not permitted from this program operator application.

DSPLYWLD=YES
Specifies that wildcards are permitted from this program operator application when the DSPLYWLD start option is FULLWILD or POAONLY.

EAS

```
►►─┬─EAS=509──────────────────┬──────────────────────────────────────────►◄
   │                          │
   └─EAS=number_of_sessions───┘
```

statements: *GROUP, APPL*
dependencies: *none*
range: *0–65535*

Specifies the estimated number of concurrent sessions this application program will have with other logical units (LU-LU sessions). VTAM uses this operand in a search to find the representation of a session between the application program and a logical unit.

The specified value should include LU 6.2 sessions and non-LU 6.2 sessions.

To avoid obtaining unnecessary common service area (CSA) storage for TSO users, you should code EAS=1 when defining TSO applications to VTAM. Do not use the default.

Note: You should code EAS on applications that have only a few concurrent sessions, to conserve storage.

Accurate coding of the EAS value for your applications can save storage in your system. For example, if you estimate that there will be less than 30 sessions with this application, but you let the EAS value default to 509, then an extra 4K table will be allocated from common storage. The size of the

table is based on the EAS value that you code and is determined as follows:

Table size	EAS value
4K	30–4000
8K	4001–8000
16K	8001–16000
32K	16001–32000
64K	32001–48000
128K	48001–56000
256K	56001–64000
512K	greater than 64000

If your EAS value is specified as lower than the number of sessions that you actually have, sessions would still be established as usual. However, the efficiency of searching for the session representation could be impaired if a smaller table was allocated due to the lower EAS value.

ENCR

```
►►─┬─ENCR=OPT──────────┬──────────────────────────────────────────►◄
   │                   │
   └─ENCR=─┬─OPT──┬────┘
           ├─COND─┤
           ├─NONE─┤
           ├─REQD─┤
           └─SEL──┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies whether this application program has any special requirements for enciphering and deciphering messages.

ENCR=COND
Specifies that this application program will establish an encrypted session if the session partner supports encryption and cryptographic product support is available. If support is not available at session start, a clear session (without encryption) is established.

VTAM starts sessions in the clear when:

- ENCR=COND is coded on the APPL definition of the PLU,
- The SLU has an APPL definition with ENCR=COND or an LU definition with ENCR=OPT
- A cryptograpic support is unavailable.

There is no attempt to start data encryption in the middle of a clear session when the encryption product becomes available again.

In VTAM, the COND option includes a test to determine if cryptograpic product support is available on the primary logical unit (PLU). However, if either end of a session specifies a higher level of encryption than ENCR=COND (ENCR=SEL

or ENCR=REQD), an encrypted session is required and no test is made to determine if cryptographic product support is available.

ENCR=NONE
Specifies that this application program cannot have cryptographic sessions.

ENCR=OPT
Specifies that the application program has no special cryptographic requirements; its cryptographic capability is the same as the host processor's capability.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from the application program. If ENCR=REQD, this application program cannot be started unless the host with which it is associated can handle cryptographic sessions. ENCR=REQD also means that this application program cannot start a session with a logical unit that cannot encipher and decipher its messages.

ENCR=SEL
Specifies that this application program can choose which messages are enciphered by VTAM. ENCR=SEL also means that this application program cannot start a session with a logical unit that cannot encipher and decipher its messages. If you code ENCR=SEL, this application program cannot be started unless the host in which it resides can handle cryptographic sessions.

ENCRTYPE

```
►►─┬─ENCRTYPE=DES───┬──────────────────────────────────────────►◄
   └─ENCRTYPE=TDES24─┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

Note: The ENCRTYPE value coded on the GROUP will be sifted down to the appropriate resource (APPL).

The SYSDEFed ENCRTYPE of an APPL resource can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to “MODEENT Macroinstruction” on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements:	*GROUP, APPL*
dependencies:	*TSO sessions only*

Determines how session establishment is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The primary logical unit (PLU) acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU’s application program, only a single bind is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) which establish sessions with TSO.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST PASS macroinstruction with the PASS option as part of their session establishment procedure.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

HAVAIL

```
►►─┬─HAVAIL=NO─────────────┬──────────────────────────────────────────►◄
   │                       │
   └─HAVAIL=─┬─NO──┬─────┘
             └─YES─┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Determines whether the application can support extended recovery facility (XRF) sessions.

HAVAIL=NO
Specifies that this application program cannot have an XRF session.

HAVAIL=YES
Specifies that this application program can have an XRF session, can initiate an XRF session, or can be the primary logical unit in an XRF session that is initiated by a secondary logical unit.

LIMQSINT

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └LIMQSINT=timeout_value────┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*
range:	*0–65535 seconds*

Specifies how often VTAM searches the queue for unused LU 6.2 contention winners.

If LIMQSINT=0, contention-winner LU 6.2 limited resource sessions are terminated immediately after their conversations end.

As a starting point, *timeout_value* should be at least one second less than half of the shortest time cost interval. For example, if the time cost interval for a line is 120 seconds, you should specify 59 seconds. Because applications can work differently, you might need to adjust the time cost interval as necessary.

Sessions that are active for longer than the *timeout_value* seconds are deactivated on the expiration of the timer and the next queue search.

LMDENT

```
    ┌LMDENT=19──────────────────┐
►►──┼───────────────────────────┼──────────────────────────────────────────►◄
    └LMDENT=number_of_entries───┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*
range:	*0–65535*

Specifies the number of entries to be used for this application program's hash table of remote logical units. VTAM uses the hash table to identify and locate remote logical units based on their LU names.

The value you specify should take into account both the largest possible number of remote logical units and the degree of efficiency in accessing the remote logical unit definitions.

LOSTERM

```
   ┌─LOSTERM=NORMAL──────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   │              ┌─NORMAL─┐ │
   └─LOSTERM=─────┤        ├─┘
                  ├─IMMED──┤
                  └─SECOND─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies how VTAM terminates a session with a hung LU. You can force an LU to an inactive state when it becomes hung during a session with an application program that has a LOSTERM exit routine but no NSEXIT exit routine.

LOSTERM=IMMED
VTAM sends an UNBIND to break the session on the first call to the LOSTERM exit.

LOSTERM=NORMAL
VTAM does not terminate the session.

LOSTERM=SECOND
VTAM allows the LOSTERM exit to issue a CLSDST on the first call to the LOSTERM exit. If a second call is made, VTAM sends an UNBIND to break the session.

LUAPFX

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └─LUAPFX=string────┘
```

statements: *APPL*
dependencies: *none*

Specifies the first two characters of the LUALIAS assigned when a dynamically generated cross-network CDRSC (with NQNMODE=NQNAME) is created for a session with the application. VTAM concatenates the characters specified with the next sequential number available to form a VTAM-generated LUALIAS name for the cross-network dynamic CDRSC.

LUAPFX=string
Indicates the two characters to be used as the prefix for all dynamically generated LUALIAS names for dynamic cross-network CDRSCs in session with this application. You must take into account the VTAM naming conventions when choosing this prefix. See “Format of Definition Statements” on page 9 for naming conventions.

By using this prefix you can eliminate the need to predefine a CDRSC with LUALIAS for each cross-network resource which may want to establish a session with an application that does not support network qualified names. See “LUALIAS” on page 104 for a description of the operand and how it is used.

Note: If model definitions are being used to define multinode persistent application programs and if PERSIST=MULTI is specified, use an MVS system symbol as part of the LUAPFX value to reduce name collisions during recovery processing.

MAC

```
►►─┬─MAC=NONE──────────┬──────────────────────────────────────────►◄
   │                   │
   │          ┌─NONE─┐ │
   └─MAC=─────┼─COND─┼─┘
              └─REQD─┘
```

statements:	*GROUP, APPL*
dependencies	*none*

Specifies the level of message authentication support required for this application program.

MAC=COND
Specifies that the application is capable of engaging in message supported sessions, and that MACing is required if the session partner is MAC capable.

MAC=NONE
Specifies that this application program is not capable of engaging in message authentication supported sessions.

MAC=REQD
Specifies that VTAM must use a message authentication code on all messages to and from this application program.

Note: A mandatory message authentication support and cryptographic enhancements session is established only under the following conditions:

- The owing host is capable of handling message authentication sessions.
- The session partner is capable of supporting message authentication.

MACLNTH

```
►►─┬─MACLNTH=4─────────┬──────────────────────────────────────────►◄
   │                   │
   │          ┌─4─┐    │
   └─MACLNTH=─┼─2─┼────┘
              │─6─│
              └─8─┘
```

statements:	*GROUP, APPL*
dependencies	*none*

Specifies the minimum length, in bytes, of the message authentication code that is to be generated.

MACLNTH is based on MACTYPE, and can be one of the following values:

MACTYPE	MACLNTH
CRC	2 or 4
DES	4, 6, or 8

MACTYPE

```
►►─┬─MACTYPE=CRC──────────┬──────────────────────────────────────────►◄
   │                      │
   └─MACTYPE=─┬─CRC─┬─────┘
              └─DES─┘
```

statements:	*GROUP, APPL*
dependencies	*none*

Specifies the type of message authentication support and cryptographic enhancements code checking VTAM will use to create or to verify the message authentication code.

MACTYPE=CRC
Specifies that VTAM use an internal cyclic redundancy check (CRC) algorithm to generate a message authentication code.

MACTYPE=DES
Specifies that VTAM use the data encryption standard (DES) services provided by the available Common Cryptographic Architecture (CCA) product to generate a message authentication code for this application.

Note: ENCRTYPE affects the type of encryption used when calculating message authentication code for DES or TDES24.

MAXPVT

```
►►─┬─MAXPVT=0──────────────────┬──────────────────────────────────────►◄
   │                           │
   └─MAXPVT=number_of_bytes────┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Specifies how much VTAM can use of the private storage allocated to the application program's task. In most cases, each task is associated with only one application program. If a task involves more than one application program, the effective MAXPVT value for the task is the sum of the values specified for all the application programs opened under the task.

number_of_bytes can be expressed as a single number up to 8 digits, or up to 7 digits with M or K. The limit for MAXPVT is approximately 2048M (2 gigabytes).

For CMIP services application programs, the minimum value is 2 megabytes.

MAXPVT=0
Specifies that no limit is defined and thus, when no storage is available in the address space associated with this application program, the session is cleared.

MAXPVT=*number_of_bytes*

MAXPVT=*number_of_bytes***K**
number_of_bytes The number of one-kilobyte segments that this application program task can use of its private area (address space). *number_of_bytes* is a decimal integer. Its value is rounded up to the next multiple of 4.

For CMIP services application programs, if the number of bytes specified is fewer than 2 megabytes, the value is rounded up to 2 megabytes.

MAXPVT=*number_of_bytes***M**
number_of_bytes The maximum amount of private area storage, in one-megabyte increments, that the application program task can use. *number_of_bytes* is a decimal integer.

For CMIP services application programs, if the number of bytes specified is less than 2 megabytes, the value is rounded up to 2 megabytes.

MDLENT

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name─┘
```

statements: *GROUP, APPL*
dependencies: *MDLTAB*

Specifies the name of the model name table entry to be used for this application program.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB.

MDLTAB

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─────┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies the name of the model name table to be used for this application program. See “Model Name Table” on page 720 for a description of how to define a model name table.

If you omit the MDLTAB operand, VTAM will not provide the name of the model name table of the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. Refer to “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MODETAB

```
┌─MODETAB=ISTINCLM──────────────┐
►►──┤                             ├──────────────────────────────────────────►◄
└─MODETAB=logon_mode_table_name─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Specifies the logon mode table to be used to associate each logon mode name with a set of session parameters for the application program if the application program participates as the secondary end of the session (SLU). The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the application program, on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

MODSRCH

```
┌─MODSRCH=NEVER──────────┐
►►──┤                       ├──────────────────────────────────────────►◄
│              ┌─NEVER─┐  │
└─MODSRCH=─┤              ├─┘
│              ├─FIRST─┤
└─LAST──┘
```

statements: *GROUP, APPL*
dependencies: *Model application definition*

Specifies when the INQUIRE APPSTAT and INQUIRE STATUS macroinstructions will search for this model definition. The VTAM containing the application that issued the INQUIRE first searches its tables for a previously defined application. If one is not found, it searches for a model that can satisfy the request.

Note: If you plan to use the TSO/VTAM BASENAME parameter, then you must code MODSRCH=FIRST on the TSO application definition statement which represents the subordinate TSO APPLs. For more information on MODSRCH, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide.*

MODSRCH=FIRST
Specifies that VTAM search for a model definition before searching the network for the specified name. For example, because the OPEN ACB macroinstruction can use a model definition, you could specify MODSRCH=FIRST for an application issuing INQUIRE APPSTAT to determine if an OPEN ACB will follow.

MODSRCH=LAST
Specifies that VTAM search for a model definition only if all other hosts in the network report that the application was not found.

MODSRCH=NEVER
Specifies that VTAM search only for the name specified on the INQUIRE macroinstruction.

name

```
►►──┬──────┬──────────────────────────────►◄  ►►──name──────────────────────────────►◄
    └name──┘
```

statements: VBUILD *statements:* GROUP, APPL

Assigns a name to the resource being defined by the definition statement. The name is used as the network name for the resource.

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On GROUP: *name* is required.

On APPL: *name* is required. It specifies a minor node name for this application program, as does the ACBNAME operand. This name, when used with the network identifier (NETID as specified on the VTAM START command) is the network-qualified name of the application program, in the form of *netid.name*. This name is sometimes called the real name to distinguish it from an alias or USERVAR name. This name must be unique in the following situations:

- Within the domain in which the application resides
- Throughout the network if the application program participates in cross-domain sessions
- Throughout all networks if the application program participates in cross-network sessions, unless network-qualified names, LUALIAS, or the alias translation function is used.

name must follow the rules for naming application programs as described in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and can include the following wildcard characters:

Asterisk (*)
: Represents 0 or more unspecified characters. An asterisk can be used as the second to eighth character.

Question mark (?)
: Represents a single unspecified character. A question mark can be used as the first to eighth character.

Restrictions: You cannot use wildcard characters to define a model application program definition in the following situations:

- If you code the ACBNAME operand on the APPL definition statement
- If you are defining a TSO/VTAM application program (AUTH=TSO on the APPL definition statement). Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about defining TSO/VTAM application programs.

For information about how to use wildcard characters to create model application program definitions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

A logical unit can request a session with an application by using one of the following names associated with the application:

- An application program network name, the *name* operand (from any domain in the network)
- The ACB name associated with the application (from within the same domain).

A logical unit using the application program's network name (the *name* operand on the APPL definition statement) can establish a session with the application program from any domain in the network. A logical unit using the ACB name to establish a session with an application program must be in the same domain.

If you do not code ACBNAME, this name must be the same as the name in the APPLID operand of the ACB macroinstruction for this application program. *OS/390 eNetwork Communications Server: SNA Programming* describes the ACB macroinstruction.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for considerations when coding the name to define terminal control address space (TCAS) or a time sharing option (TSO) user.

Note: If you duplicate names for application program minor nodes, it is not advisable to register more than one of the duplicate names at a time. If there are duplicate application program major node names and more than one is registered, you will receive messages about the duplication.

OPERCNOS

```
►►─┬─OPERCNOS=NALLOW──────────┬──────────────────────────────────────────►◄
   │                          │
   └─OPERCNOS=─┬─NALLOW─┬─────┘
               └─ALLOW──┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*

Specifies whether the operator can issue the MODIFY CNOS and MODIFY DEFINE commands.

The MODIFY CNOS command changes the CNOS parameters governing LU 6.2 sessions. The MODIFY DEFINE command establishes new entries or modifies existing entries in the LU-mode table for the specified application program.

OPERCNOS=ALLOW
Allows the operator to issue the MODIFY CNOS command to change the CNOS parameters governing LU 6.2 sessions.

OPERCNOS=NALLOW
Prevents the operator from issuing the MODIFY CNOS command to change the CNOS parameters governing LU 6.2 sessions.

Note: OPERCNOS applies only to the modification of CNOS parameters. Therefore, it does not affect the operator's ability to display CNOS information.

PARSESS

```
                 (2)
  ┌─PARSESS=YES──────┐
                 (1)
  ├─PARSESS=NO───────┤
  ►►──────────────────────────────────────────────────────►◄
                 ┌─NO──┐
  └─PARSESS=─────┴─YES─┘
```

Notes:

1 PARSESS defaults to NO when APPC=NO.

2 PARSESS defaults to YES when APPC=YES.

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*

Specifies whether this application program can have multiple concurrent (parallel) sessions with another application program or independent logical unit in the same domain or in another domain.

PARSESS=NO
Restricts this application program to one LU-LU session, between itself and another logical unit, at a time. VTAM uses netnames rather than CIDs if PARSESS=NO.

PARSESS=YES
Allows this application program to have multiple LU-LU sessions between the same session partners.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of parallel sessions in a multiple network environment.

PERSIST

```
  ┌─PERSIST=SINGLE───────┐
  ►►──────────────────────────────────────────────────────►◄
                 ┌─SINGLE─┐
  └─PERSIST=─────┴─MULTI──┘
```

statements:	*APPL*
dependencies:	*none*
range:	*N/A*

Specifies whether the application is capable of multinode persistent sessions.

PERSIST=SINGLE
Indicates single-node persistence only is allowed. SINGLE can be abbreviated as S.

PERSIST=MULTI

Indicates multinode persistence is allowed in addition to single-node persistence support. MULTI can be abbreviated as M.

Notes:

1. VTAM will issue error message `IST1350I` if PERSIST=MULTI is coded and the application program is activated in an environment that does not support multinode persistent sessions. Refer to *OS/390 SecureWay Communications Server: IP and SNA Codes*, *OS/390 SecureWay Communications Server: IP Messages Volume 1 (EZA)*, and *OS/390 SecureWay Communications Server: IP Messages Volume 2 (EZB)* for details.
2. When coded at a Network Node (or ICN), PERSIST=MULTI is allowed only on a model application definition.
3. If model definitions are being used to define multinode persistent application programs and if LUAPFX is specified on the model definition statement, use an MVS system symbol as part of the LUAPFX value to reduce name collision during recovery processing.

POAQLIM

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─POAQLIM=number_of_messages───┘
```

statements:	*GROUP, APPL*
dependencies:	*AUTH=PPO or AUTH=SPO*
range:	*1–2147483647*

Specifies the number of messages that can be queued for this application program. If you do not code POAQLIM, no limit is used.

POAQLIM applies only to a program operator application program (AUTH=PPO or AUTH=SPO).

Code POAQLIM high enough so the limit will be reached only if there is an apparent error. A limit of five times the number of tasks using the program operator application (POA) times the number of lines in the longest expected message group should be enough to capture all necessary data without using excessive amounts of storage.

The limit for the PPO should be somewhat higher than the SPO to allow for unsolicited messages and PPOLOG messages.

It is also recommended that you increase VTAM private storage by the sum of all the POAQLIM values specified in your system times 152 bytes, if possible. This is done by increasing the REGION parameter on the VTAM start procedure.

PRTCT

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─PRTCT=password───────────────┘
```

statements:	*GROUP, APPL*

dependencies: *none*

Specifies a 1–8 character password that verifies this application program's authority to run. VTAM compares the password to the one in the application program's access method control block (ACB) when an OPEN macroinstruction is issued. If you do not code this operand, no password verification is done.

For TSO applications, you must use the same password for TCAS and for each TSO/VTAM user.

REGISTER

```
      ┌─REGISTER=CDSERVR──────┐
►►────┼───────────────────────┼──────────────────────────────────────────►◄
      │                       │
      │              ┌─CDSERVR─┐
      └─REGISTER=────┼─────────┼──┘
                     ├─NETSRVR─┤
                     └─NO──────┘
```

statements: *GROUP, APPL*
dependencies: *APPN only*

Specifies how a resource should be registered.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to the network node server, but that directory resource registration should not be requested for it.

REGISTER=NO
Indicates that the resource should not be registered.

Note: Because TSO subtasks are not required to have network-unique names, only TSO main tasks should be registered and REGISTER=NO should be coded on the definitions of all subtask application programs.

SECACPT

```
      ┌─SECACPT=NONE──────────┐
►►────┼───────────────────────┼──────────────────────────────────────────►◄
      │              ┌─NONE────┐
      └─SECACPT=─────┼─────────┼──┘
                     ├─ALREADYV┤
                     ├─AVPV────┤
                     ├─CONV────┤
                     └─PERSISTV┘
```

statements: *GROUP, APPL*
dependencies: *APPC=YES*

Defines whether LU 6.2 conversation requests containing access security subfields are accepted by this application's logical unit.

SECACPT=ALREADYV
Specifies that this logical unit supports conversation requests containing access security subfields. The logical unit also accepts the already-verified indications that it receives in conversation requests from partner LUs.

SECACPT=AVPV
Specifies that this logical unit supports conversation requests containing access security subfields. The logical unit also accepts the already-verified indications and the persistent verification indications that it receives in conversation requests.

SECACPT=CONV
Specifies that this logical unit supports conversation requests containing access security subfields.

SECACPT=NONE
Specifies that this logical unit does not support conversation requests containing access security subfields.

SECACPT=PERSISTV
Specifies that this logical unit supports conversation requests containing access security subfields. The logical unit also accepts the persistent verification indications that it receives in conversation requests.

SECLVL

```
                  ┌─SECLVL=ADAPT──────┐
►►──────────────────┼──────────────────┼──────────────────────────────────────►◄
                  │                   │
                  │          ┌─ADAPT──┐│
                  └─SECLVL=──┼─────────┼┘
                             ├─LEVEL1──┤
                             └─LEVEL2──┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*

Specifies whether VTAM uses the basic or enhanced security protocol when performing session-level LU-LU verification. SECLVL specifies the type of verification; the related VERIFY operand specifies whether VTAM performs verification.

This operand is valid only when APPC=YES and session-level LU-LU verification is used. (VERIFY=OPTIONAL or VERIFY=REQUIRED must be coded on this application program major node. If you code VERIFY=NONE or let it default, a parameter conflict results.)

SECLVL=ADAPT
Specifies that either the enhanced or the basic protocol is used for sessions that use session-level LU-LU verification. VTAM attempts to use the enhanced protocol but accepts the use of the basic protocol if a partner LU does not support the enhanced protocol.

SECLVL=LEVEL1
Specifies that the basic protocol is used for sessions that use session-level

LU-LU verification. If a partner LU does not support the basic protocol, VTAM rejects the session and issues a sense code.

SECLVL=LEVEL2
Specifies that VTAM uses only the enhanced protocol for sessions that use session-level LU-LU verification. If a partner LU does not support the enhanced protocol, VTAM rejects the session and issues a sense code.

SESSLIM

```
►►─┬─SESSLIM=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─SESSLIM=─┬─NO──┬────┘
              └─YES─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Determines whether this application program has a session limit between it and another application program, usually an SLU, which handles one session at a time.

SESSLIM=NO
Specifies that this application program does not have a session limit.

SESSLIM=YES
Specifies that this application program has a session limit of one.

SONSCIP

```
►►─┬─SONSCIP=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─SONSCIP=─┬─NO──┬────┘
              └─YES─┘
```

statements: *GROUP, APPL*
dependencies: *none*

Determines whether the application program receives UNBIND RUs in its SCIP exit routine when the application program is acting as the primary end of a failing session.

SONSCIP=NO
VTAM schedules the application program's NSEXIT routine with a CLEANUP RU. If the application program does not have an NSEXIT routine, VTAM schedules the LOSTERM exit routine.

SONSCIP=YES
VTAM ends the session for the application program (no CLSDST need be issued). VTAM schedules the application program's SCIP exit routine with an UNBIND RU that contains information (type codes) describing the failure. For a description of these type codes, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

Note: Because secondary application programs receive UNBIND RUs in their SCIP exit routines, it is not necessary for these application programs to code SONSCIP=YES to receive the UNBIND type codes.

SPAN

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └─SPAN=span_name───┘
```

statements:	*GROUP, APPL*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SRBEXIT

```
    ┌─SRBEXIT=NO──────┐
►►──┼─────────────────┼──────────────────────────────────────────────►◄
    │            ┌─NO─┐│
    └─SRBEXIT=───┼────┼┘
                 └─YES┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Specifies whether this application program is authorized to use service request block (SRB) processing in its exit routines.

Note: If multiple applications open an ACB under one task, the SRBEXIT operand is recognized for the first application opening the ACB. Therefore, the first open ACB determines whether the exit routines execute in SRB or TCB mode. The SRBEXIT operand is ignored for all subsequent applications opening the ACBs under the same task, If SRBEXIT=YES is specified on the first application opening the ACB, the subsequent applications opening ACBs under the same task must be authorized.

SRBEXIT=NO
Specifies that the application program is not authorized to use SRB processing in its exit routines.

SRBEXIT=YES
Specifies that all the application program's exit routines are branch-entered in SRB mode, supervisor state, key 0. An application must be APF-authorized, key 0–7, or in supervisor state to open an ACB whose corresponding APPL definition statement has SRBEXIT=YES coded.

SSCPFM

```
   ┌─SSCPFM=USSPOI──────┐
►►─┼────────────────────┼──────────────────────────────────────────►◄
   │            ┌─USSPOI─┐│
   └─SSCPFM=────┼────────┼┘
                └─USSNOP─┘
```

statements: *GROUP, APPL*
dependencies: *AUTH=PPO or AUTH=SPO*

Specifies the default USS table that is used to send a VTAM operator message to the program operator if the message is not included in a user-defined table. SSCPFM has no effect on the USS table that is used to process VTAM operator commands.

SSCPFM applies only to a program operator application program (AUTH=PPO or AUTH=SPO). It is ignored if you code it for an application program that is not a program operator.

SSCPFM=USSNOP
Specifies that ISTINCNO is the name of the default USS table to be used for VTAM messages sent to this program operator. Note that ISTINCNO is used only for messages that are not included in a user-defined table.

SSCPFM=USSPOI
Specifies that ISTCFCMM is the name of the default message table to be used for VTAM messages sent to this program operator. Note that ISTCFCMM is used only for messages that are not included in a user-defined table. After VTAM searches ISTCFCMM, it searches ISTINCNO. This value is recommended if you run command lists.

SYNCLVL

```
   ┌─SYNCLVL=CONFIRM────┐
►►─┼────────────────────┼──────────────────────────────────────────►◄
   │            ┌─CONFIRM┐│
   └─SYNCLVL=───┼────────┼┘
                │─NONE───┼
                └─SYNCPT─┘
```

statements: *GROUP, APPL*
dependencies: *APPC=YES*

Specifies the application program's synchronization support level.

SYNCLVL=CONFIRM
Specifies that synchronization levels of NONE and CONFIRM are supported by this application program. This logical unit supports conversation-specified confirmation processing, but not sync point processing.

SYNCLVL=NONE
Specifies that the application program does not support conversation-specified confirmation processing or sync point processing.

SYNCLVL=SYNCPT
Specifies that synchronization levels of SYNCPT, CONFIRM, and NONE are supported. This logical unit supports both conversation-specified confirmation and sync point processing.

TYPE

```
►►──TYPE=APPL──────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Specifies that this VBUILD definition statement defines an application program major node.

USSTAB

```
                ┌─USSTAB=ISTINCNO──────────┐
►►──────────────┼──────────────────────────┼──────────────────────►◄
                └─USSTAB=uss_table_name────┘
```

statements:	*GROUP, APPL*
dependencies:	*AUTH=PPO or AUTH=SPO*

Specifies a user-defined USS table that contains user-modified VTAM operator commands and VTAM operator messages to be used by the program operator.

USSTAB applies only to a program operator application program (AUTH=PPO or AUTH=SPO). It is ignored if you code it for an application program that is not a program operator.

If there is no USSTAB operand in the definition for a program operator:

- VTAM uses the IBM-supplied USS table (ISTINCNO) when processing USS commands issued by the program operator.
- VTAM uses the SSCPFM operand to determine which USS table to use when sending a VTAM operator message to a program operator, as described in the section on the SSCPFM operand.

VCNS

```
                ┌─VCNS=NO──────────────────┐
►►──────────────┼──────────────────────────┼──────────────────────►◄
                │          ┌─NO──┐         │
                └─VCNS=────┼─────┼─────────┘
                           └─YES─┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Specifies whether VTAM is allowed to perform VTAM Common Network Services (VCNS) functions.

Note: If you use VTAM Common Network Services, you must code VCNS=YES.

VCNS=NO
Specifies that VCNS functions are not allowed.

VCNS=YES
Specifies that VCNS functions are allowed.

VERIFY

```
►►─┬─VERIFY=NONE──────────┬──────────────────────────────────────────►◄
   │                      │
   └─VERIFY=─┬─NONE────┬──┘
             │         │
             ├─OPTIONAL┤
             │         │
             └─REQUIRED┘
```

statements:	*GROUP, APPL*
dependencies:	*APPC=YES*

Specifies whether VTAM performs session-level LU-LU verification during activation of LU-LU 6.2 sessions. The related SECLVL options specify the type of verification. If you do not code SECLVL, VTAM attempts to use the enhanced protocol if VERIFY=OPTIONAL or VERIFY=REQUIRED is specified, but will accept basic protocol.

If you code VERIFY=OPTIONAL or VERIFY=REQUIRED, a security management product, such as RACF®* 1.9 or later, must be installed and active, and a profile for the LU needs to be in the security management product's database. With RACF 1.9, the APPCLU class needs to be active and a profile of the LU needs to be in the APPCLU class; if not, the OPEN ACB will fail. If the application specifies PARMS=(NQNAMES=YES) on the ACB, the profile for the LU must include the remote partner LU's network identifier. Refer to the *OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide* for more information.

VERIFY=NONE
Specifies that no verification of the partner LU's identity takes place during session activation.

Note that a parameter conflict results if you code SECLVL on the same application program major node as VERIFY=NONE.

VERIFY=OPTIONAL
Specifies that identity verification is performed for certain partner LUs during session activation. Determination for which partner LUs the LU-LU verification is performed depends on whether there is a password defined for the LU-LU pair in the installed security manager product. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about how VTAM determines when to perform LU-LU verification.

VERIFY=REQUIRED
Specifies that VTAM verifies the identity of all partner LUs during activation of sessions between LU 6.2 applications. Every partner LU must have an LU-LU password defined. Any partner LUs that do not have an LU-LU password defined cannot establish LU 6.2 sessions with this application program.

VPACING

```
►►─┬─VPACING=7──────────────────┬──────────────────────────────────────►◄
   └─VPACING=number_of_requests─┘
```

statements:	*GROUP, APPL*
dependencies:	*none*
range:	*0–63*

Specifies the maximum number of normal-flow requests that another logical unit can send to this application program in a session before waiting to receive a pacing response. This value, which controls the pacing of requests to the application program, is exchanged when a session is established.

If the value is greater than 63, the default value is used. No pacing of requests to the application program is done if VPACING=0.

Note: Subarea VTAM and NCP support sessions between dependent LUs that are not paced. However, APPN sessions require pacing, so if you do not specify a value for VPACING, VTAM uses its default to ensure pacing.

The value of VPACING can be expressed with leading zeros and is limited to eight digits.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

VTAMFRR

```
►►─┬─VTAMFRR=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─VTAMFRR=─┬─NO──┬────┘
              └─YES─┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Determines whether a VTAM functional recovery routine (FRR) takes effect when control is returned to the application program from an SRB mode applications program interface (API) request or when an SRB exit routine is scheduled.

VTAMFRR=NO
VTAM returns control after purging any VTAM FRRs.

VTAMFRR=YES
A VTAM FRR always takes effect when the application program:

- Receives control back from an SRB-authorized path application program interface (API) request
- Receives control in an SRB exit routine

Channel-Attachment Major Node

To define a channel-attachment major node, code a VBUILD definition statement for the major node and lower-level definition statements for each minor node.

The channel-attachment major node defines the following types of support:
- Channel-to-channel adapter (CTCA)
- Multipath channel (MPC)
- Channel-attached NCP

Each link group has a unique GROUP definition statement and contains a unique set of minor node statements. The operands and default values also vary according to the particular link group being defined.

Channel-to-Channel Adapter Support

To define channel-to-channel adapter (CTCA) support, code the following definition statements:

```
VBUILD
GROUP LNCTL=CTCA
LINE
PU
```

VBUILD Definition Statement

See “VBUILD Definition Statement” on page 70 for details on the VBUILD definition statement.

GROUP Definition Statement

Code a GROUP definition statement to define a channel-to-channel line group.

LINE Definition Statement

Code one LINE definition statement for each channel-to-channel adapter. The LINE definition statement defines the characteristics of the line to VTAM.

PU Definition Statement

Code one PU definition statement for each LINE definition statement. VTAM uses the PU definition statement to define the link station for an adjacent host processor.

Table 5. Channel-Attachment (CTCA) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	80	R			
		CONFGDS	73				
		CONFGPW	73				CONFGDS
name	GROUP	LNCTL=CTCA	74	R			
		REPLYTO	79			3.0	MIH
		SPAN	79				NetView
name	LINE	ADDRESS	72				
		ISTATUS	74		G	ACTIVE	

Table 5. Channel-Attachment (CTCA) Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		MAXBFRU	75		G	1	
		MIH	76		G	NO	LNCTL=SDLC
		SPAN	79				NetView
name	PU	DELAY	73		G,L	0	
		ISTATUS	74		G	ACTIVE	
		PUTYPE	78		G,L	4	
		SPAN	79				NetView
		TGN	80			1	PUTYPE=4, 5

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Multipath Channel Support

To define multipath channel (MPC) support, code the following definition statements:

VBUILD
GROUP LNCTL=MPC
LINE
PU

VBUILD Definition Statement

See “VBUILD Definition Statement” on page 70 for details on the VBUILD definition statement.

GROUP Definition Statement

Code a GROUP definition statement to define a multi-path channel connection between two hosts.

LINE Definition Statement

Code only one LINE definition statement for multi-path channel support. If you code more than one LINE definition statement for the group, the activation will fail for the entire group. The LINE definition statement defines the characteristics of the line to VTAM.

PU Definition Statement

Code one PU definition statement for the LINE definition statement. VTAM uses the PU definition statement to define the link station for an adjacent host processor.

Table 6. Channel-Attachment (MPC) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	80	R			
		CONFGDS	73				
		CONFGPW	73				CONFGDS
name	GROUP	LNCTL=MPC	74	R			
		REPLYTO	79			3.0	
		SPAN	79				NetView
name	LINE	READ	78	R			LNCTL=MPC
		WRITE	80	R			LNCTL=MPC
		ISTATUS	74		G	ACTIVE	
		MAXBFRU	75		G	1	
		SPAN	79				NetView
name	PU	ISTATUS	74		G	ACTIVE	
		PUTYPE	78		G,L	4	
		SPAN	79				NetView
		TGN	80			1	PUTYPE=4, 5

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Channel-Attached NCP Support

To define a channel-attached NCP, code the following definition statements:

VBUILD
GROUP LNCTL=NCP
LINE
PU

VBUILD Definition Statement

See “VBUILD Definition Statement” on page 70 for details on the VBUILD definition statement.

GROUP Definition Statement

Code a GROUP definition statement to define channel-to-NCP link groups. If you define a line in more than one line group, only one group can be active at a time.

LINE Definition Statement

Code one LINE definition statement for each channel-to-NCP link. The LINE definition statement defines the following characteristics to VTAM:

- The name for the link

- The device number for the link
- Certain procedural options to be used for this line

PU Definition Statement

Code one PU definition for each LINE definition statement. VTAM uses the PU definition statement to define the link station for the adjacent communication controller.

You can define a channel attachment between a host and an NCP in a channel-attachment major node. This data host attachment eliminates the requirement that the host have ownership (control) of network resources. With this channel attachment defined, VTAM can contact a channel-attached NCP without activating the NCP.

When you define a channel attachment between a host and an NCP, be sure to consider the following:

- Providing a data host to back up a communication management configuration (CMC) host
- Coding the PCCU definition statement in a CMC
- Coding the HOST definition statement in a CMC

For more information on these considerations, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Coding the PCCU Definition Statement

Several operands on the PCCU definition statement are used to define channel characteristics. If you do not define a channel in a channel-attachment major node, the values on the PCCU definition statement are used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node, the definition statements in the channel-attachment major node contains operands similar to those on the PCCU definition statements.

For a data host, the CHANCON, MAXDATA, and DELAY operands on the PCCU definition statement are replaced by the same operands on the PU definition statement in the channel-attachment major node. In addition, the CUADDR operand on the PCCU definition statement is replaced by the ADDRESS operand on the LINE definition statement in the channel-attachment major node.

Activating the channel-attachment major node to activate the channel causes VTAM to use the values defined in the channel-attachment major node. Later, if the NCP is activated, these same operands, if they appear on the NCP's PCCU definition statements, are ignored. If you define the channel in a channel-attachment major node and would like to avoid confusion concerning these operands, avoid coding duplicate values on the PCCU definition statement.

The RNAME operand on the PCCU definition statement can name a link station associated with a channel link defined in a channel-attachment major node. When the host activates each NCP, the link station and link defined in the channel-attachment major node representing the channel are activated automatically.

Coding the HOST Definition Statement

The HOST definition statement in the NCP major node has a MAXBFRU operand for defining the number of buffers that the host can use for receiving data from the NCP.

If you do not define the channel in a channel-attachment major node, the MAXBFRU value on the HOST definition statement is used because the channel is implicitly defined and activated automatically when the NCP is activated.

However, when you define the channel in a channel-attachment major node,

- The MAXBFRU operand on the LINE definition statement for that major node defines the number of buffers the host can use for receiving data.
- VTAM ignores the MAXBFRU value on the HOST definition statement when the link station for the channel is activated automatically. This is because the value is determined by the RNAME operand when the NCP is activated.

Recommendation: The MAXBFRU operand on the LINE definition statement of the CA definition should contain the same value as the MAXBFRU operand specified on the HOST definition statement in the NCP major node definition.

Table 7. Channel-Attachment (NCP) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	80	R			
		CONFGDS	73				
		CONFGPW	73				CONFGDS
name	GROUP	LNCTL=NCP	74	R			
		SPAN	79				NetView
name	LINE	ADDRESS	72				
		ISTATUS	74		G	ACTIVE	
		MAXBFRU	75		G	3	
		SPAN	79				NetView
name	PU	CHANCON	72		G,L	COND	LNCTL=NCP
		DELAY	73		G,L	0	
		ISTATUS	74		G	ACTIVE	
		MAXDATA	76		G,L	65535	LNCTL=NCP
		PUTYPE	78		G,L	4	
		SPAN	79				NetView
		TGN	80			1	PUTYPE=4, 5

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code one VBUILD definition statement for each channel-attachment major node.

You can file the channel-attachment definition statements for the different types of support as one major node or as separate major nodes. Use operational considerations such as activation processing overhead or switched-network backup capabilities to determine a practical number of major nodes.

If multiple support definition statements are contained in a major node, code only one VBUILD definition statement for the major node.

Basic Channel-to-Channel Mode

For channel-to-channel adapter and multipath channel definitions, the ESCON® basic channel-to-channel (BCTC) mode is not supported. Specify serial channel-to-channel (SCTC) mode instead.

Full Syntax

The full syntax for the channel-attachment major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=CA──────────────────────────────────►
    └─name─────┘

►──┬──────────────────────────────────────────────────────────────┬──►◄
   ├─CONFGDS=confgds_name──────────────────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────────────────┘
```

GROUP

```
                                   ┌─REPLYTO=3.0──────────┐
►►──name──GROUP──LNCTL=──┬─CTCA─┬──┤                      ├──────────────►
                         ├─NCP──┤  └─REPLYTO=timeout_value┘
                         └─MPC──┘

►──┬──────────────┬──────────────────────────────────────────────────►◄
   └─SPAN=span_name┘
```

LINE

```
►►──name──LINE──┬──────────────────────────┬──READ=(─channel_unit_address──)──────►
                └─ADDRESS=device_address───┘
```

```
►─WRITE=(─channel_unit_address──)──┬─ISTATUS=ACTIVE──────────────────┬──►
                                   │                                 │
                                   └─ISTATUS=──┬─ACTIVE──┬──────────┘
                                               └─INACTIVE─┘

                   (2)
   ┌─MAXBFRU=3──────────────┐
                   (1)
   ├─MAXBFRU=1──────────────┤   ┌─MIH=NO──────┐
   │                        │   │             │
►──┤                        ├───┤             ├──────┬──────────────────┬──►◄
   │                        │   │             │      │                  │
   └─MAXBFRU=number_of_buffers─┘ └─MIH=──┬─NO──┬─┘      └─SPAN=span_name─┘
                                         └─YES─┘
```

Notes:

1 MAXBFRU defaults to 1 when LNCTL=CTCA or LNCTL=MPC.

2 MAXBFRU defaults to 3 when LNCTL=NCP.

PU

```
                                                   (2)
                                              ┌─DELAY=0──────────────┐
                                                   (1)
                 ┌─CHANCON=COND──────┐        ├─DELAY=0──────────────┤
►►──name──PU─────┤                   ├────────┤                      ├──────────────────────────────►
                 │                   │        │                      │
                 └─CHANCON=──┬─COND──┬─┘        └─DELAY=timeout_value─┘
                             └─UNCOND─┘

   ┌─ISTATUS=ACTIVE──────────────┐   ┌─MAXDATA=65535──────────────┐
►──┤                             ├───┤                            ├──────┬──────────────┬──►
   │                             │   │                            │      │              │
   └─ISTATUS=──┬─ACTIVE──┬─────────┘   └─MAXDATA=number_of_bytes────┘      └─PUTYPE=4────┘
               └─INACTIVE─┘

   ┌─TGN=1──────────────┐
►──┤                    ├──────────────────────────────────────────────────────────────────────►◄
   │                    │
   └─SPAN=span_name─────┘   └─TGN=──┬─1──────┬──────────┘
                                    ├─ANY────┤
                                    └─tg_number─┘
```

Notes:

1 DELAY defaults to 0 when LNCTL=NCP is coded.

2 DELAY defaults to 0 when LNCTL=CTCA is coded.

Operand Descriptions

The following section describes the operands you can code for the channel-attachment major node.

ADDRESS

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─ADDRESS=device_address───┘
```

statements: LINE
dependencies: none

Specifies the three- or four-digit device number of the channel-to-channel adapter (CTCA) or the channel-attached NCP. If you do not code this operand, the VTAM operator must supply the device number by specifying the U operand of the VARY ACT command. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for details on using the U operand.

Note: For a channel-attached NCP, this channel-link definition might conflict with a channel-link name automatically defined by VTAM if you attempt to have this channel definition and an automatically defined channel link for the same device number active at the same time. (A channel link is defined automatically if an NCP is activated and a CUADDR value is included on its PCCU definition statement, or if the operator issues a VARY ACT command with U=*device number*.) The second activation (either this definition or the automatic definition) will fail. For more information on channel-attached NCPs, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

CHANCON

```
►►──┬─CHANCON=COND──────────┬──────────────────────────────────────────►◄
    │                       │
    └─CHANCON=──┬─COND──┬───┘
                └─UNCOND─┘
```

statements: *GROUP, LINE, PU*
dependencies: *LNCTL=NCP*

Determines whether VTAM sends a conditional or an unconditional channel Contact request to the NCP when the host is to activate an NCP over a channel.

CHANCON=COND
VTAM allows the NCP to reject the host's Contact request if the host's subarea number is already in use by another host (attached to another of the NCP's channels). This results when duplicate subareas are assigned to more than one host (attached to more than one channel interface). For instance, two hosts might have inadvertently been assigned the same number, or they might have been assigned duplicate numbers to allow mutual backup capabilities.

CHANCON=UNCOND
VTAM sets an indicator in the channel Contact request that forces the NCP to accept the Contact request regardless of whether a host with that subarea number is currently channel-attached to that NCP. If another host using the same subarea number is channel-attached to the NCP, the NCP breaks contact with the host on the other channel.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └CONFGDS=name──────┘
```

statements: *VBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Put a data definition (DD) statement that has the CONFGDS name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └CONFGPW=password──┘
```

statements: *VBUILD*
dependencies: *CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

DELAY

For LNCTL=CTCA:		**For LNCTL=NCP:**	
`►►─┬─DELAY=0──────────────┬──►◄` `└─DELAY=timeout_value─┘`		`►►─┬─DELAY=0──────────────┬──►◄` `└─DELAY=timeout_value─┘`	
statements:	GROUP, LINE, PU	*statements:*	GROUP, LINE, PU
dependency:	none	*dependency:*	none
range:	0-9.999 seconds	*range:*	0-9.999 seconds

Specifies the maximum amount of time, in seconds and thousandths of seconds, that VTAM waits before transmitting low-priority data (path information unit (PIU) transmission priority 0 or 1) to the physical unit.

For more efficient operation, code this value as high as possible. For maximum speed across the interface, set this value to 0. Refer to the description of the DELAY operand in a channel-to-channel-attached host processor in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information.

ISTATUS

```
►►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────────►◄
   │                         │
   └─ISTATUS=─┬─ACTIVE───┬───┘
              └─INACTIVE─┘
```

statements: *CLUSTER, GROUP, LINE, PU, TERMINAL*
dependencies: *none*

Specifies whether the resource is to become active when its major node is activated.

ISTATUS=ACTIVE
The resource becomes active when its major node becomes active.

ISTATUS=INACTIVE
The resource remains inactive until the operator activates it.

The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ACT command.

LNCTL

```
►►─LNCTL=─┬─CTCA─┬──────────────────────────────────────────────────────►◄
          ├─NCP──┤
          └─MPC──┘
```

statements: *GROUP*
dependencies: *none*

Specifies the type of line support being defined.

LNCTL=CTCA
Specifies that the lines are channel-to-channel attachment lines.

LNCTL=MPC
Specifies that the lines are multipath channel-attachment lines.

LNCTL=NCP
Specifies that the lines are channel-to-NCP lines.

When you code LNCTL=NCP, it is not necessary for VTAM to record the resources attached to the NCP.

MAXBFRU

For CTCA, MPC:

```
                 (2)
      ┌MAXBFRU=5──────────────┐
                 (1)
      ├MAXBFRU=1──────────────┤
►►────┼───────────────────────┼──────────►◄
      └MAXBFRU=number_of_buffers┘
```

For NCP:

```
      ┌MAXBFRU=3──────────────────────┐
►►────┼───────────────────────────────┼──────►◄
      └MAXBFRU=number_of_buffers──────┘
```

Notes:

1 MAXBFRU defaults to 1 when HPR=NO, XID=NO, CONNTYPE=LEN, or the VTAM node is not HPR capable.

2 MAXBFRU defaults to 5 when HPR=YES, XID=YES, CONNTYPE=APPN, and the VTAM node is HPR capable.

statements:	GROUP, LINE	statements:	GROUP, LINE
dependency:	none	dependency:	none
range:	1-16	range:	1-255

Specifies the amount of storage VTAM uses to receive data from the channel-attached resource once it's activated. The value you code for this operand depends on the type of channel support you are defining.

CTCA and MPC

The value coded is the number of 4K-buffer pages. The resulting buffer size is *number_of_pages* multiplied by 4K. Make sure that the buffer size is large enough to contain the largest anticipated data transfer.

VTAM automatically substitutes a value of 16 for any coded value larger than 16 without issuing a warning message.

CTCA

If the channel-to-channel adapter is communicating with a pre-V3R2 host, MAXBFRU specifies the number of buffers (rather than pages) to be used.

Some devices might have a hardware restriction on the maximum number of buffers, but VTAM does not verify the value for those devices. Refer to the appropriate publication for the device before coding this operand.

MPC

The same MAXBFRU value is used for each of the READ subchannels. The resulting buffer size is *number_of_pages* multiplied by 4K. The total buffer space used is 4K times *number_of_pages*, times the number of READ subchannels.

If there is a hardware buffer size for MPC, VTAM compares the MAXBFRU value and the hardware buffer size and uses the smaller of the two to avoid wasting fixed storage. MAXBFRU is not used to determine the buffer size of the WRITE

subchannels coded on this host. The WRITE subchannel buffer size is determined dynamically by VTAM depending on the setting at the READ end of the channel.

You can reduce this value after examining VTAM multipath channel tuning statistics so that excessive storage is not allocated for each subchannel. To obtain tuning statistics output, code the TNSTAT start option. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on MPC tuning statistics.

NCP

Specifies the number of buffers to be allocated by the host for receiving data from an NCP. This operand has the same meaning as the MAXBFRU operand on the HOST definition statement in the NCP major node.

The value coded for MAXBFRU must satisfy the condition that MAXBFRU times the IOBUF size is less than 65535.

MAXDATA

```
►►─┬─MAXDATA=65535──────────────┬──────────────────────────────────────────►◄
   │                            │
   └─MAXDATA=number_of_bytes────┘
```

statements:	*GROUP,LINE,PU*
dependencies:	*LNCTL=NCP*
range:	*5–65535*

Specifies the maximum number of bytes that the NCP can receive in one segment of a path information unit (PIU), including the transmission header (TH) and the request or response header (RH).

If a MODIFY LOAD command will be issued for a local or remote NCP, the size of MAXDATA must be at least 2082 (2048 plus 34, the size of the PIU header).

Note: If you are not defining the channel to the NCP in a channel-attachment major node, code the MAXDATA operand on the PCCU definition statement. Otherwise, code it on the PU definition statement.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about choosing a value for MAXDATA.

MIH

```
►►─┬─MIH=NO──────────┬──────────────────────────────────────────────────►◄
   │                 │
   └─MIH=─┬─NO──┬────┘
          └─YES─┘
```

statements:	*GROUP,LINE*
dependencies:	*LNCTL=CTCA*

Causes the link to become inoperative if a start I/O timeout occurs because the receiving host is in disabled compatibility mode. MIH stands for missing-interrupt handler. The default time value associated with this operand is 3 minutes, although you can change the default with the MIHTMOUT start option.

MIH=NO
The MIH default of NO means that VTAM does not deactivate the link if REPLYTO plus MIHTMOUT time units expire without receipt of an interrupt.

MIH=YES
If MIH=YES is coded on a CTCA LINE statement (or sifts down to it from the GROUP statement), VTAM deactivates the link if the following time elapses after START I/O is issued from the link: REPLYTO (from the GROUP statement, default 3 seconds) plus MIHTMOUT (start option, default 3 minutes), and no interrupt has yet been received.

Note: If you do not code MIH, a link can appear active but is most likely hung or unusable. Therefore, coding MIH=YES is recommended.

name

statements:	VBUILD	*statements:*	CLUSTER, GROUP, LINE, LU, PU, TERMINAL

Assigns the name to the resource being defined. The name is used as the network name for network resources.

On VBUILD

name is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On CLUSTER

name is required. It provides the minor node name of the cluster controller represented by this definition statement.

On GROUP

name is required. It provides the minor node name of the line group represented by this definition statement.

On LINE

name is required. It provides the minor node name of the line represented by this statement.

On LU

name is required. It provides the minor node name of the logical unit represented by this definition statement.

On PU

name is required. It provides the minor node name of the physical unit represented by this definition statement.

On TERMINAL

name is required. It provides the minor node name of the logical unit (terminal) represented by this definition statement.

PUTYPE

```
►►──┬──────────┬──────────────────────────────────────────────────────►◄
    └PUTYPE=4──┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*none*

Specifies the physical unit type. PUTYPE=4 is the only valid value.

For an MPC node, when any other value is coded, an error message is issued and the value is changed to the default value of 4.

READ

```
►►──READ=(─channel_unit_address──)──────────────────────────────────►◄
```

statements:	*LINE*
dependencies:	*LNCTL=MPC*
range:	*X'000'–X'FFFF'*

Specifies the hexadecimal channel_unit_addresses used to read data from the adjacent host. You must code at least one READ channel_unit_address. If you code an address that is not defined to the system, the activation will fail for the entire group.

Channel_unit_addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen. VTAM generates addresses for the range. For example, READ=(132, 128-130, 13A) will define the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A.

For each address on the READ operand, you must code the corresponding address on the WRITE operand in the adjacent host to provide a complete path.

Do not code the same channel_unit_address for the READ operand and the WRITE operand in the same host.

REPLYTO

```
►►─┬─REPLYTO=3.0──────────────┬──────────────────────────────────────────►◄
   └─REPLYTO=timeout_value────┘
```

statements:	*GROUP*
dependencies:	*MIH*
range:	*.1–25.5*

Specifies the amount of time VTAM will wait for completion of an operation. If this timeout expires during normal operation, a message is written to inform the operator that a timeout has occurred.

A 3.0 seconds *timeout_value* should be adequate unless the attached resource is disabled for long intervals.

CTCA

Specifies the number of seconds that VTAM waits for completion after starting a channel program.

A STATION INOP can be scheduled at this time if a missing interrupt handler (MIH) purge has already taken place. The differences between the MIH and REPLYTO operands are:

- The REPLYTO operand causes a message to be generated if a channel program is not completed after the time specified (or 3 seconds, if no time is specified).
- The MIH=YES operand causes the link to become inoperative, defaulting to 3 minutes after the start I/O, if a timeout occurs because of the receiving host.

MPC

Specifies how long VTAM waits for completion of an MPC XID I/O operation after starting a channel program. After the XID completes, REPLYTO has no meaning.

SPAN

```
►►─┬──────────────────────────┬──────────────────────────────────────────►◄
   └─SPAN=span_name───────────┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

TGN

```
►►─┬─TGN=1──────────────┬──────────────────────────────────────────►◄
   │                    │
   └─TGN=─┬─1──────────┬─┘
          ├─ANY────────┤
          └─tg_number──┘
```

statements:	*PU*
dependencies:	*PUTYPE=4 or 5*
range:	*1–255*

Assigns a transmission group (TG) number that represents the channel connecting two subareas. The TG number (assigned to the same channel) coded in the two subareas must match.

TGN=*tg_number*
Specifies a decimal number assigned to the transmission group.

TGN=ANY
Allows the adjacent subarea node to accept whatever valid transmission group number it receives on the exchange identifier (XID). You can specify TGN=ANY only for the PU at one end of a transmission group. The PU at the other end must have a valid transmission group number or default to 1. An error occurs if you specify TGN=ANY for both PUs.

TYPE

```
►►─TYPE=CA──────────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Means that the VBUILD definition statement defines a channel-attachment major node to VTAM.

WRITE

```
►►─WRITE=(─channel_unit_address──)──────────────────────────────────►◄
```

statements:	*LINE*
dependencies:	*LNCTL=MPC*
range:	*X'000'–X'FFFF'*

Specifies the hexadecimal channel_unit_addresses used to write data to the adjacent host. If you specify an address that is offline or not defined to the system, You must code at least one WRITE channel_unit_address.

Channel_unit_addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen. VTAM generates addresses

for the range. For example, WRITE=(132, 128-130, 13A) will define the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A.

For each address on the WRITE operand, you must code the corresponding address on the READ operand in the adjacent host to provide a complete path.

Do not code the same channel_unit_address for the READ operand and the WRITE operand in the same host.

The WRITE channel_unit_address are used in order, so certain channel_unit_address can be given a higher priority by coding them first. For example, you might want to code channel_unit_address with a higher bandwidth first.

Cross-Domain Resource (CDRSC) Major Node

The cross-domain resource (CDRSC) major node is used to define:
- Resources in another domain
- Independent LUs

To define one or more cross-domain resource (CDRSC) major nodes, code a VBUILD definition statement, an optional GROUP definition statement, and a CDRSC definition statement for each minor node.

In an SNA-interconnected network, define a CDRSC major node with a VBUILD definition statement, optional NETWORK definition statements, and one or more CDRSC definition statements.

If VTAM is part of a multiple-domain network, two definitions with the same resource name (a logical unit and a CDRSC) can co-exist. In a backup and recovery situation where one host is assuming ownership of a logical unit from another host in the same network, the logical unit can be activated by the new host even though an application program within it currently has a cross-domain session with the logical unit. If the physical units and logical units being recovered support ACTPU(ERP) and ACTLU(ERP) requests, the sessions with the logical units are not affected when the physical units and logical units are activated. The CDRSC definition automatically becomes a shadow resource and the logical unit is now defined as a same-domain (APPL, LU, or LOCAL) resource. If the current host relinquishes ownership of the logical unit, it does so by releasing or deactivating the logical unit and its associated physical unit, causing VTAM to activate its CDRSC again, making it a cross-domain resource.

Independent LU

There are several ways to define independent LUs,
- Dynamically define CDRSCs
- Statically define CDRSCs
- Statically define LUs (LOCADDR=0)

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for detailed information on the methods of defining independent LUs. This section contains a description of the definition statements for statically defining an independent LU through the CDRSC definition statement.

Registering Independent LUs

Because the characteristics of an independent LU and of the node on which it resides are not recorded by VTAM, it is difficult to determine the appropriate level of registration for all cases. Therefore, independent LUs are not registered by default because registration of the same resource from multiple nodes can result in the generation of alerts in the APPN network or sub-optimal routing of LU-LU sessions.

However, if an independent LU is owned by a LEN node (that is not acting as an end node in the network), the registration of the independent LU is recommended. Because LEN nodes cannot participate in APPN searches, their LUs cannot be found unless the LUs are defined to an APPN node connected to the LEN node. In this case, you can code the REGISTER operand on the CDRSC or LU definition to register the independent LU to the network node server, or to the network node server and the central directory server. If a large group of independent LUs requires registration, the REGISTER operand can be specified on a higher level definition statement to take advantage of the sift-down effect.

Note: Do not specify the REGISTER operand on CDRSC statements on which the CPNAME operand has been coded because the CPNAME operand indicates that the independent LU resides on another APPN node that will be responsible for registering the LU. If the two operands are coded on the same CDRSC statement, either explicitly or by the sift-down effect, the REGISTER operand will default to NO and be ignored.

If you dynamically change an independent LU, the rules cited in “Rules for Multiple Definition of Resources” on page 83 are followed.

APPN Cross-Domain Resources

Defining APPN cross-domain resources is optional, and it is recommended that you not define them to save time. However, there are cases where you might need to define APPN cross-domain resources. For instance, if you want APPN and LEN connectivity to a given independent LU, then both APPN and LEN adjacent link stations need to be defined in the independent LU’s adjacent link station list.

In an APPN network, you can define APPN cross-domain resources only at network nodes. If you define APPN cross-domain resources at an end node, an error message indicates that the APPN information is not valid.

To define an APPN cross-domain resource, code the CPNAME operand on the CDRSC definition statement. When you code CPNAME, the APPN directory is updated with the location of the cross-domain resource. However, if the resource is found in the directory to have been registered or dynamically defined, the update is ignored, but no message is generated.

The CPNAME and NETSRVR operands are both valid on the CDRSC definition statement. If both operands have the same value, the resource specified by CPNAME is assumed to be the network node. If CPNAME and NETSRVR have different values, the resource specified by NETSRVR is assumed to be the network node.

If only CPNAME is defined under the CDRSC definition statement, the resource it specifies is assumed to be a network node. If the resource is not a network node, an error message indicates that the APPN information is not valid.

The CPNAME and CDRM operands can be specified on the same CDRSC definition statement to allow a path to be defined through the subarea and APPN networks to reach the resource.

Rules for Multiple Definition of Resources

The connection of an independent LU (CDRSC) to an adjacent link station is only for the duration of the independent LU's sessions. In addition, the independent LU can have multiple concurrent sessions using different adjacent link stations.

The rules for multiple definition of resources are presented in Table 8. This table assumes that a CDRSC is currently defined to VTAM (existing resource). It also assumes that the operator issues a command to activate a major node that contains new definition statements (input resource). Multiple definition of resources occurs when the name of an input resource matches the name of an existing resource.

Table 8. Rules for Multiple Definition of Resources

Existing Resource	Input Resource	Major node after integration	Notes™
Dynamic CDRSC	Dynamic CDRSC	Not applicable	1
Dynamic CDRSC	Predefined Independent LU	ISTPDILU	2, 5
Dynamic CDRSC	Predefined CDRSC	CDRSC major node	2, 5
Predefined Independent LU	Dynamic CDRSC	Not applicable	1
Predefined Independent LU	Predefined Independent LU	Not applicable	4
Predefined Independent LU	Predefined CDRSC	ISTPDILU	2, 6
Predefined CDRSC	Dynamic CDRSC	Not applicable	1
Predefined CDRSC	Predefined Independent LU	ISTPDILU	3, 7
Predefined CDRSC	Predefined CDRSC	Not applicable	4
Predefined APPL	Predefined CDRSC	CDRSC major node	5
Predefined LU	Predefined CDRSC	CDRSC major node	5
MNPS application	Predefined CDRSC	Not applicable	8

Notes:

1. This case does not apply to multiple definition of resources. When a resource currently exists, no new dynamic CDRSC is created or considered.
2. If the two resource definitions have different NETID values, both are specified with NQNMODE=NAME, either by definition or start option, and the existing resource has active or pending sessions, the input resource is not defined to VTAM. If the existing resource does not have any sessions, the input resource's NETID value is also the resulting resource's NETID value.

   If the two resource definitions have different NETID values and at least one is specified with NQNMODE=NQNAME, either by definition or start option, the resource definitions are not merged.
3. If the two resource definitions have different NETID values and both are specified with NQNMODE=NAME, either by definition or start option, the input

resource is not defined to VTAM. This is true whether the existing resource has sessions or not. If the NETID values of the two resource definitions do not conflict, the input resource's NETID value is also the resulting resource's NETID value.

If the two resource definitions have different NETID values and at least one is specified with NQNMODE=NQNAME, either by definition or start option, the resource definitions are not merged.

4. If both are specified with NQNMODE=NAME, either by definition or start option, a duplicate definition of the resource occurs, and the input resource is not defined to VTAM.

   If the two resource definitions have different NETID values and at least one is specified with NQNMODE=NQNAME, either by definition or start option, the resource definitions are not merged.
5. All operands from the input resource apply to the existing resource.
6. CDRSC operands from the input resource apply to the existing resource.
7. Only LU operands from the input resource apply to the existing resource.
8. Predefining a CDRSC at a network node will fail if an MNPS application is already active using the same name. `IST489I` is issued.

Exceptions to the merging of multiple resource definitions might occur. These exceptions occur because of different NETID values for the existing and new resource definition, which are both specified with NQNMODE=NAME (either by definition or start option). Changes in the NETID of a given resource can occur only if the resource does not have any existing or pending sessions.

If either the existing resource or the input resource is specified with NQNMODE=NQNAME and they have different NETID values, the resource definitions are not merged.

Merging multiple resource definitions causes operands from the input resource to overlay values in the existing resource only in cases where the operands of the existing resource are default values. Defined values on operands of the existing resource are not overlaid with default values on operands of the input resource. Default values and sifted values from the input resource apply to the existing resource definition as well, so long as they do not override defined values on operands of the existing resource. However, not all operands are necessarily integrated as a result of the second resource definition.

The method used for determining what operands to integrate from the input definition to the existing definition is as follows:

- If the existing resource is a dynamic or predefined CDRSC and the input resource is a predefined independent LU, the following operands on the CDRSC definition statement are integrated:
  - ASLENT
  - ASLTAB
  - DLOGMOD
  - EAS
  - ENCR
  - ENCRTYPE
  - LOGAPPL
  - LUALIAS [3]

3. LUALIAS is only integrated in some cases. See Table 9 on page 85.

  - MDLENT
  - MDLTAB
  - MODETAB
  - NETID
  - PACING
  - RESSCB
  - VPACING
- If the existing resource is a predefined independent LU and the input resource is a predefined CDRSC, the following operands on the CDRSC definition statement are integrated:
  - ALSREQ
  - CDRM
  - ALSLIST [4]
  - LUALIAS [3]
  - NETID
  - NQNMODE [5]
  - VFYOWNER
- If the existing resource is a predefined CDRSC, the input resource is a predefined CDRSC, they have different NETID values, and both are specified with NQNMODE=NAME, either by definition or start option, the second definition fails. (See note 2 on page 83.)

  If the two resource definitions have different NETID values and at least one is specified with NQNMODE=NQNAME, either by definition or start option, the resource definitions are not merged.
- If the existing resource is a member of ISTPDILU, the input resource is a predefined independent LU, they have different NETID values, and both are specified with NQNMODE=NAME, either by definition or start option, all operands on the LU definition statement are integrated.

  If the two resource definitions have different NETID values and at least one is specified with NQNMODE=NQNAME, either by definition or start option, the resource definitions are not merged.

When multiple resource definitions are merged, the resulting values for LUALIAS and NQNMODE are determined by several factors. The following tables show when LUALIAS and NQNMODE values are integrated.

LUALIAS Values after a Merge of Resource Definitions

If the existing resource is in session and has an LUALIAS name defined to it, the LUALIAS name cannot be removed or changed as a result of a resource definition merge. However, when a resource definition merge occurs because the input resource is a predefined CDRSC without NETID specified, the LUALIAS is removed. See Table 9 for specific examples.

Table 9. Examples of How LUALIAS Values Are Merged

Existing Resource	LUALIAS	Session	Input Resource	LUALIAS	Notes
Predefined Independent LU	N	No	Predefined CDRSC	Y	2

4. The list of adjacent link stations from the input resource and the existing resource are always combined.

5. NQNMODE is only integrated in some cases. See “NQNMODE Values after a Merge of Resource Definitions” on page 87.

Table 9. Examples of How LUALIAS Values Are Merged (continued)

Existing Resource	LUALIAS	Session	Input Resource	LUALIAS	Notes
Predefined Independent LU	N	Yes	Predefined CDRSC	Y	2
Predefined Independent LU	Y	No	Predefined CDRSC	Y	2, 3
Predefined Independent LU	Y	No	Predefined CDRSC	N	3, 4
Predefined Independent LU	Y	Yes	Predefined CDRSC	Y	1
Predefined Independent LU	Y	Yes	Predefined CDRSC	N	1
Predefined Independent LU	S	Yes	Predefined CDRSC	S	5
Predefined CDRSC	Y	No	Predefined Independent LU	N	2
Predefined CDRSC	Y	Yes	Predefined Independent LU	N	2
Predefined Independent LU	Y	Yes	Predefined CDRSC	PWN	1
Predefined Independent LU	Y	No	Predefined CDRSC	PWN	3, 4
Predefined CDRSC	Y	Yes	Predefined Independent LU	PWN	1
Predefined CDRSC	Y	No	Predefined Independent LU	PWN	6, 7

Note:

1. The resource definitions are not merged because the existing resource is in session.
2. The resource definitions are merged, and the LUALIAS of the CDRSC applies to the ILU.
3. The LUALIAS of the ILU is removed.
4. An LUALIAS is no longer associated with the ILU.
5. The resource definitions are merged.
6. An LUALIAS is no longer associated with the CDRSC.
7. The LUALIAS of the CDRSC is removed.

Table 9. Examples of How LUALIAS Values Are Merged (continued)

Existing Resource	LUALIAS	Session	Input Resource	LUALIAS	Notes
Symbols in LUALIAS column: Y: Resource has an LUALIAS, that is different from the LUALIAS of the resource it is being integrated with. N: Resource does not have an LUALIAS; either LUALIAS was not specified or it was specified with no *lualias* value. S: Resource has an LUALIAS that is the same as the LUALIAS of the resource it is being integrated with. PWN: The resource is predefined without a network identifier, so LUALIAS is not valid.					

NQNMODE Values after a Merge of Resource Definitions

NQNMODE values after merges of resources are explained in the following tables:

- Table 10
- Table 11 on page 88
- Table 12 on page 89
- Table 13 on page 90

Using the NQNMODE Merge Tables:

- In Table 10, choose the resources that are being merged (one input resource and one existing resource) by looking at the resource types and the NQNMODE status, and then choosing the corresponding resource type numbers.
- Using the two numbers (one for the input resource and one for the existing resource), look at Table 11 on page 88 for input resources 1–5, or Table 12 on page 89 for input resources 6–10, and find where the column for the input resource intersects the row for the existing resource.

  Note: Only existing resources can be dynamic, in session, applications, or LUs. So, all resource types in Table 10 are valid for the existing resource, but only resource types 1–10 are valid for the input resource.
- The intersection has a letter followed by numbers in parentheses. These letters and numbers correspond to notes about the merge. These notes are listed in Table 13 on page 90.

Table 10. Resource Types for NQNMODE Merges. This table shows the resource types that are involved in the merge. Information includes whether the resource has sessions, and the current value of NQNMODE. NQNMODE must be either NAME or NQNAME.

Resource Type	NQNMODE	Defined with NETID	Sessions
1 - Predefined CDRSC	NQNAME defaulted	Yes	No
2 - Predefined CDRSC	NQNAME defined	Yes	No
3 - Predefined CDRSC	NAME defaulted	Yes	No
4 - Predefined CDRSC	NAME defined	Yes	No
5 - Predefined CDRSC	n/a	No	No
6 - Predefined Independent LU	NQNAME defaulted	Yes	No
7 - Predefined Independent LU	NQNAME defined	Yes	No
8 - Predefined Independent LU	NAME defaulted	Yes	No

Cross-Domain Resource Major Node

Table 10. Resource Types for NQNMODE Merges (continued). This table shows the resource types that are involved in the merge. Information includes whether the resource has sessions, and the current value of NQNMODE. NQNMODE must be either NAME or NQNAME.

Resource Type	NQNMODE	Defined with NETID	Sessions
9 - Predefined Independent LU	NAME defined	Yes	No
10 - Predefined Independent LU	n/a	No	No
11 - Predefined CDRSC	NQNAME defaulted	Yes	Yes
12 - Predefined CDRSC	NQNAME defined	Yes	Yes
13 - Predefined CDRSC	NAME defaulted	Yes	Yes
14 - Predefined CDRSC	NAME defined	Yes	Yes
15 - Predefined CDRSC	n/a	No	Yes
16 - Predefined Independent LU	NQNAME defaulted	Yes	Yes
17 - Predefined Independent LU	NQNAME defined	Yes	Yes
18 - Predefined Independent LU	NAME defaulted	Yes	Yes
19 - Predefined Independent LU	NAME defined	Yes	Yes
20 - Predefined Independent LU	n/a	No	Yes
21 - Dynamic CDRSC	NAME	n/a	No
22 - Dynamic CDRSC	NQNAME	n/a	No
23 - Dynamic CDRSC	NAME	n/a	Yes
24 - Dynamic CDRSC	NQNAME	n/a	Yes
25 - Application	n/a	n/a	No
26 - Application	n/a	n/a	Yes
27 - LU	n/a	n/a	No
28 - LU	n/a	n/a	Yes

Table 11. Matrix for NQNMODE Merges, Part 1. In this table, input resources 1–5 are read left to right across the top of the five columns. The existing resources are read top to bottom, in the far left column. Letters and numbers at the intersection of a particular input resource with an existing resource are explained in Table 13 on page 90.

EXIST. RES.	INPUT RESOURCE 1	INPUT RESOURCE 2	INPUT RESOURCE 3	INPUT RESOURCE 4	INPUT RESOURCE 5
1	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
2	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
3	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
4	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
5	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
6	**M**(2 5 12 18)	**M**(2 5 12 14 18)	**M**(1 3 12 18)	**M**(1 3 12 18 14)	**M**(11 15 18 16)
7	**M**(2 5 12 18 14)	**M**(2 5 12 18)	**M**(2 5 12 18 6)	**M**(1 3 12 18)	**M**(11 15 18 16)
8	**M**(2 4 18)	**M**(2 4 18 14)	**M**(1 5 18) **F**(20)	**M**(1 5 18 14) **F**(20)	**M**(11 18 16)
9	**M**(1 5 18 14 6)	**M**(2 4 18 14)	**M**(1 5 18) **F**(20)	**M**(1 5 18 14) **F**(20)	**M**(11 18 16)
10	**M**(2 4 23 22)	**M**(2 4 23 22)	**M**(1 5 22)	**M**(1 5 22)	**M**(11 16 5)
11	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
12	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
13	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)

Table 11. Matrix for NQNMODE Merges, Part 1 (continued). In this table, input resources 1–5 are read left to right across the top of the five columns. The existing resources are read top to bottom, in the far left column. Letters and numbers at the intersection of a particular input resource with an existing resource are explained in Table 13 on page 90.

EXIST. RES.	INPUT RESOURCE 1	INPUT RESOURCE 2	INPUT RESOURCE 3	INPUT RESOURCE 4	INPUT RESOURCE 5
14	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
15	**F**(8)	**F**(8)	**F**(8)	**F**(8)	**F**(8)
16	**M**(2 5 12 18)	**M**(2 5 12 14 18)	**M**(1 3 12 18)	**M**(1 3 12 14 18)	**M**(11 15 16 18)
17	**M**(2 5 12 13)	**M**(2 5 12 18)	**M**(2 5 6 12 13)	**F**(6 9)	**M**(11 15 16 18)
18	**M**(2 4 12 18)	**M**(2 4 12 18 14)	**M**(1 5 18) **F**(20)	**M**(1 5 18 14) **F**(20)	**M**(11 16 18)
19	**M**(1 5 6 12 13)	**F**(6 9)	**M**(1 5 18 14) **F**(20)	**M**(1 5 18 14) **F**(20)	**M**(11 16 18)
20	**M**(2 4 17 18)	**M**(2 4 17 18 14)	**M**(2 4 17 18)	**M**(2 4 17 18 14)	**M**(11 18)
21	**M**(2 4 7 12 18 21)	**M**(2 4 7 12 18 21,14)	**M**(5 18 21) **F**(20)	**M**(5 18 21) **F**(20)	**M**(7 11 15 16 18 21)
22	**M**(5 18 21)	**M**(5 18 21)	**M**(1 3 7 12 18 21)	**M**(1 3 7 12 18 21,14)	**M**(7 11 15 16 18 21)
23	**M**(2 4 7 12 18 21)	**M**(2 4 7 12 18 21,14)	**M**(5 18 21) **F**(20)	**M**(5 18 21) **F**(20)	**M**(7 11 15 16 18 21)
24	**M**(5 18 21)	**M**(5 18 21)	**M**(1 3 7 12 18 21)	**M**(1 3 7 12 18 21,14)	**M**(7 11 15 16 18 21)
25	**S**	**S**	**S**	**S**	**S**
26	**S**	**S**	**S**	**S**	**S**
27	**S**	**S**	**S**	**S**	**S**
28	**S**	**S**	**S**	**S**	**S**

Table 12. Matrix for NQNMODE Merges, Part 2. In this table, input resources are read left to right 6–10 across the top of the five columns. The existing resources are read top to bottom, in the far left column. Letters and numbers at the intersection of a particular input resource with an existing resource are explained in Table 13 on page 90.

EXIST. RES.	INPUT RESOURCE 6	INPUT RESOURCE 7	INPUT RESOURCE 8	INPUT RESOURCE 9	INPUT RESOURCE 10
1	**M**(2 5 12 13)	**F**(19)	**M**(1 3 12 13)	**F**(19)	**M**(22 15 13 17 2)
2	**M**(2 5 12 13 14)	**F**(19)	**M**(2 5 12 13 6)	**F**(19)	**M**(2 6 15 13 17 22)
3	**M**(2 4 13)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19) F(20)	**M**(1 22 13 6 17)
4	**M**(1 5 13 14 6)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19)	**M**(1 22 13 6 17)
5	**M**(2 4 13 17)	**F**(19)	**M**(1 5 13 17)	**F**(19)	**M**(11 5 13 16)
6	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
7	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
8	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
9	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
10	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
11	**M**(2 5 12 13)	**F**(19)	**M**(1 3 12 13)	**F**(19)	**M**(22 15 13 17 2)
12	**M**(2 5 12 13)	**F**(19)	**M**(2 5 6 12 13)	**F**(19)	**M**(2 6 15 13 17 22)

Cross-Domain Resource Major Node

Table 12. Matrix for NQNMODE Merges, Part 2 (continued). In this table, input resources are read left to right 6–10 across the top of the five columns. The existing resources are read top to bottom, in the far left column. Letters and numbers at the intersection of a particular input resource with an existing resource are explained in Table 13.

EXIST. RES.	INPUT RESOURCE 6	INPUT RESOURCE 7	INPUT RESOURCE 8	INPUT RESOURCE 9	INPUT RESOURCE 10
13	**M**(2 4 12 13)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19)	**M**(1 22 13 6 17)
14	**M**(1 5 6 12 13)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19)	**M**(1 22 13 6 17)
15	**M**(2 4 12 13 17)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19)	**M**(11 5 13 16)
16	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
17	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
18	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
19	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
20	**F**(8)	**F**(19)	**F**(8)	**F**(19)	**F**(8)
21	**M**(2 4 7 12 13 21)	**F**(19)	**M**(5 13 21) **F**(20)	**F**(19)	**M**(7 11 15 16 13 21)
22	**M**(5 13 21)	**F**(19)	**M**(1 3 7 12 13 21)	**F**(19)	**M**(7 11 15 16 13 21)
23	**M**(2 4 7 12 13 21)	**F**(19)	**M**(5 13 21) **F**(20)	**F**(19)	**M**(7 11 15 16 13 21)
24	**M**(5 13 21)	**F**(19)	**M**(1 3 7 12 13 21)	**F**(19)	**M**(7 11 15 16 13 21)
25	**F**(24)	**F**(19)	**F**(24)	**F**(19)	**F**(24)
26	**F**(24)	**F**(19)	**F**(24)	**F**(19)	**F**(24)
27	**S**	**F**(19)	**S**	**F**(19)	**S**
28	**S**	**F**(19)	**S**	**F**(19)	**S**

Table 13. Notes for NQNMODE Merges Tables. This table explains the letters and numbers seen in Table 11 on page 88 and Table 12 on page 89.

Letter, Number	Explanation
M	Merge the resources.
F	Fail the integration request. Do not merge the resources. Existing resource remains unchanged.
S	Input resource is put into shadow. If a shadow already exists, the shadow is treated as the existing resource, and is merged with the input resource.
1	Resulting NQNMODE value for this resource is NAME.
2	Resulting NQNMODE value for this resource is NQNAME.
3	An SRT entry is added for this resource using the netid of this host, if the existing resource is cross net.
4	The SRT entry is deleted for this resource.
5	No SRT entry addition or deletion is needed.
6	Do not override a defined NQNMODE value with a defaulted one. Do not change the NQNMODE value.
7	When the existing resource is a dynamic CDRSC, the resources will be merged regardless of conflicting NQNMODE values, whether or not the existing CDRSC has sessions. The NQNMODE value of the existing resource becomes the NQNMODE value of the input resource.

Table 13. Notes for NQNMODE Merges Tables (continued). This table explains the letters and numbers seen in Table 11 on page 88 and Table 12 on page 89.

Letter, Number	Explanation
8	It is not valid to have duplicate predefined resource definitions of the same type (for example, two CDRSC definitions, or two ILU definitions).
9	NQNMODE conflict, and the existing resource is in session. Also, the existing resource has NQNMODE defined.
10	Conflict in two defined NQNMODE values.
11	NQNMODE cannot be defined for a predefined alias.
12	Since NQNMODE=NQNAME on one of the resources, the netids of the input and existing resource must be the same to have a collision.
13	Only ILU parameters for the existing resources are overridden by parameters from the input resources. The RDTE for the existing resources is moved (if not already there) to ISTPDILU.
14	NQNMODE is now considered a defined value for the resulting resource.
15	The existing resource must have a netid of this host's netid for there to have been a collision.
16	The resulting resource becomes a predefined alias (RCDPDALS in RCDRS is turned on).
17	RCDPDALS in RCDRS is turned off.
18	The definition parameters for the existing resources are overridden by parameters from the input resources.
19	Impossible scenario. NQNMODE can not be defined on a predefined ILU when the ILU is the input resource. NQNMODE is only defined for an ILU when the ILU has merged with a CDRSC which has NQNMODE defined.
20	Fail this integration request if the netids of the existing and input resources are different.
21	Resource becomes a predefined resource.
22	Resulting resource takes the netid of the input resource.
23	The input resource must have a netid of this host's netid for there to have been a collision.
24	The definition should be failed because there might be the case where an appl and an ILU represent the same resource.

Cross-Domain Resource Major Node

Table 14. Cross-Domain Resource Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CDRSC	112	R			
		CONFGDS	99				
		CONFGPW	99				CONFGDS
name	NETWORK	NETID	107				
name	GROUP	See the CSRSC definition statement for operands.					
cdrscname	CDRSC	ADJLIST	96		G		
		ALSLIST	96		G		ALSREQ
		ALSREQ	96		G	ALSREQ start option	
		ASLENT	97		G		ASLTAB
		ASLTAB	97		G		
		CDRM	98		G		
		CPNAME	100		G		REGISTER
		DLOGMOD	100		G		MODETAB
		EAS	101		G	256	Independent LUs
		ENCR	101		G	OPT	
		ENCRTYPE	102		G		DES
		ISTATUS	103		G	ACTIVE	
		LOGAPPL	103		G		NetView or TSO
		LUALIAS	104				
		MAXSESS	105		G		NCP-attached ILUs
		MDLENT	105		G		MDLTAB
		MDLTAB	106		G		
		MODETAB	106		G	ISTINCLM	
		NETSRVR	107		G		APPN only
		NQNMODE	108		G	NAME	see description
		PACING	109		G	1	
		REGISTER	110		G	NO	CPNAME; APPN only
		RESSCB	110		G	0	Type 2.1
		SPAN	111		G		NetView
		SRCOUNT	111		G		SRCHRED start option
		SRTIMER	111		G		SRCHRED start option
		VFYOWNER	112		G	NO	
		VPACING	113		G	2	

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each set of CDRSC definition statements.

NETWORK Definition Statement

Code a NETWORK definition statement to identify the network in which the logical units represented by the CDRSC definition statements reside. One or more NETWORK definition statements can appear anywhere after the VBUILD definition statement. If you code consecutive NETWORK definition statements (without intervening CDRSC definition statements), only the last one is used.

If no NETWORK definition statement exists before a CDRSC definition statement, the resource is considered valid for any network. Such CDRSCs can be resources within VTAM's network or alias names for resources in another network.

GROUP Definition Statement

Code a GROUP definition statement to group cross-domain resources. Each minor node in the group is defined with the values specified on the operands on the GROUP definition statement unless an operand is specified on the CDRSC definition statement.

Note: LUALIAS is not supported on the GROUP definition statement because an LUALIAS name must be unique to a CDRSC.

CDRSC Definition Statement

Code CDRSC definition statements to define CDRSC minor nodes for those cross-domain resources with which an application program or logical unit within this domain can have a session.

Notes:

1. Information on coding the CDRSC definition statement for TSO/VTAM is in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.
2. The same set of CDRSC definition statements can be used throughout the network.

Full Syntax

The full syntax for the cross-domain resource (CDRSC) major node follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=CDRSC──────────────────────────────────►
    └──name────┘
```

Cross-Domain Resource Major Node

```
►──┬──────────────────────────────────────────────────────────────────────►◄
   ├─CONFGDS=confgds_name──────────────────────────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────────────────────────┘
```

NETWORK

```
►►──┬──────────┬──NETWORK──┬──────────────────────┬──────────────────────►◄
    └─name─────┘           └─NETID=network_id─────┘
```

GROUP

```
►►──name──GROUP──────────────────────────────────────────────────────────►◄
```

CDRSC

```
►►──cdrscname──CDRSC──┬──────────────────────┬──┬──────────────────────────┬──►
                      └─ADJLIST=list_name────┘  └─ALSLIST=─(─pu_name──────)─┘

►──┬─ALSREQ=value_of_alsreq_start_option──┬──────────────────────────────►
   └─ALSREQ=──┬─NO──────────────────────┬──┘
              └─YES─────────────────────┘

►──┬──────────────────────────────────────────────────────────────────────►
   └─ASLENT=associated_lu_table_entry_name──────────────────────────────────┘

►──┬──────────────────────────────────────┬──┬──────────────────────────►
   └─ASLTAB=associated_lu_table_name──────┘  └─CDRM=cdrm_name──────────────┘

►──┬──────────────────────────────────────────────────────────────────────►
   └─CPNAME=control_point_name──────────────────────────────────────────────┘

                                                   ┌─EAS=256──────────────┐
►──┬──────────────────────────────────────────────┬──┬──────────────────┬──►
   └─DLOGMOD=default_logon_mode_table_entry_name──┘  └─EAS=number_of_sessions─┘

   ┌─ENCR=OPT──────┐  ┌─ENCRTYPE=DES──────┐  ┌─ISTATUS=ACTIVE──────────┐
►──┼────────────────┼──┼────────────────────┼──┼─────────────────────────┼──►
   └─ENCR=──┬─OPT──┬─┘  └─ENCRTYPE=TDES24───┘  └─ISTATUS=──┬─ACTIVE───┬─┘
            ├─COND─┤                                       └─INACTIVE─┘
            ├─NONE─┤
            ├─REQD─┤
            └─SEL──┘
```

```
►──┬──────────────────────────────────┬──┬──────────────────┬──────────────────────────────►
   └─LOGAPPL=controlling_primary_lu─┘  └─LUALIAS=lualias─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   └─MAXSESS=max_number_LU-LU_sessions─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   └─MDLENT=model_name_table_entry_name─┘

                                          ┌─MODETAB=ISTINCLM──────────────┐
►──┬──────────────────────────────────┬──┬──────────────────────────────────────────────►
   └─MDLTAB=model_name_table_name─┘  └─MODETAB=logon_mode_table_name─┘

►──┬──────────────────────────────────────────────────────────────────────────────────────►
   └─NETSRVR=network_node_server─┘

   ┌─NQNMODE=value_of_nqnmode_start_option──────┐
►──┼──────────────────────────────────────────────┼──────────────────────────────────────►
   └─NQNMODE=┬─value_of_nqnmode_start_option─┬─┘
             ├─NAME──────────────────────────┤
             └─NQNMODE───────────────────────┘

   ┌─PACING=1──────────────┐   ┌─REGISTER=NO──────────┐
►──┼───────────────────────┼───┼──────────────────────┼──────────────────────────────────►
   └─PACING=┬─1──────────┬─┘   └─REGISTER=┬─NO───────┬─┘
            └─pacing_number─┘               ├─CDSERVR──┤
                                            └─NETSRVR──┘

   ┌─RESSCB=0──────────────────────────────┐
►──┼───────────────────────────────────────┼──┬──────────────────┬──────────────────────►
   └─RESSCB=number_of_session_control_blocks─┘  └─SPAN=span_name─┘

►──┬──────────────────────────────────────┬──┬──────────────────────────────────┬──────►
   └─SRCOUNT=number_of_search_requests─┘  └─SRTIMER=number_of_seconds─┘

   ┌─VFYOWNER=NO──────┐   ┌─VPACING=2──────────────────────┐
►──┼──────────────────┼───┼────────────────────────────────┼──────────────────────────►◄
   └─VFYOWNER=┬─NO───┬─┘   └─VPACING=┬─2──────────────────┬─┘
              └─YES──┘               └─number_of_requests─┘
```

Operand Descriptions

The following section describes the operands you can code for the cross-domain resource (CDRSC) major node.

ADJLIST

```
►►──┬──────────────────────────┬──────────────────────────────────────────────►◄
    └─ADJLIST=list_name────────┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies the name of the ADJLIST definition statement which contains the list of adjacent SSCPs to be used for all session requests for this CDRSC.

ALSLIST

```
►►──┬──────────────────────────┬──────────────────────────────────────────────►◄
    └─ALSLIST=─(─pu_name──────)─┘
```

statements: *GROUP, CDRSC*
dependencies: *ALSREQ*

Specifies a list of adjacent link station names that VTAM can use to gain session connectivity to the independent LU you are defining. The adjacent link station might be currently defined. Validation of the names occurs during session activation.

You can include ISTAPNPU in the list instead of a real adjacent link station name. ISTAPNPU is a generic representation for an APPN adjacent link station. If VTAM finds ISTAPNPU in the adjacent link station list for an independent LU, VTAM treats ISTAPNPU as if it were a real APPN adjacent link station when trying to determine whether to attempt routing through the APPN network.

If ALSREQ=YES, all session requests for this independent LU must be received over an adjacent link station that matches one in the adjacent link station list. For a LEN adjacent link station, the names must be the same to be considered a match. For an APPN adjacent link station, if any other APPN adjacent link station is in the adjacent link station list (including ISTAPNPU) it is considered a match.

Note: For a LEN adjacent link station, the NETID must match the NETID value of the independent LU at session establishment. You are responsible for ensuring that PUs named in an adjacent link station list have NETIDs that correspond to the independent LU resource. An APPN adjacent link station has no such restriction.

There might be special ALSLIST requirements for certain migration configurations involving LEN connections. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for details.

ALSREQ

```
    ┌─ALSREQ=value_of_alsreq_start_option─┐
►►──┤                                     ├──────────────────────────────────►◄
    └─ALSREQ=──┬─NO──┬──────────────────┘
               └─YES─┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies whether adjacent link station names must match an entry in the default adjacent link station list of the independent LU at a session request.

If you do not code ALSREQ, the value coded on the ALSREQ start option is used.

ALSREQ=NO
Specifies that an adjacent link station name does not have to match an entry in the default adjacent link station list of the independent LU at a session request.

ALSREQ=YES
Specifies that an adjacent link station name must match an entry in the default adjacent link station list of the independent LU at a session request.

If the adjacent link station names are required, only session requests over an adjacent link station which matches one of the adjacent link stations in the list are allowed. An APPN adjacent link station matches any APPN adjacent link stations in the list (including ISTAPNPU). A LEN adjacent link station matches only an adjacent link station in the list that has the same name.

If adjacent link station names are not required, any adjacent link station can receive a session request for this LU. Furthermore, if the adjacent link station is LEN, its name is automatically added to the adjacent link station list for the independent LU. If the adjacent link station is APPN, the generic APPN adjacent link station name (ISTAPNPU) is automatically added to the adjacent link station list for the independent LU instead of the real adjacent link station name.

ASLENT

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─ASLENT=associated_lu_table_entry_name─┘
```

statements: *GROUP, CDRSC*
dependencies: *ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the table named in ASLTAB by default.

ASLTAB

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─ASLTAB=associated_lu_table_name─┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application program in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

CDRM

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─CDRM=cdrm_name───┘
```

statements:	*GROUP, CDRSC*
dependencies:	*none*

Specifies the name of the CDRM that controls the CDRSC. The CDRM must be in the same domain as the CDRSC.

If you do not code the CDRM operand, VTAM sends session setup requests to SSCPs in the default SSCP list, until either the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM saves the name of the owning SSCP. If the end of the list is reached, the session setup fails.

If you code CDRM, VTAM moves the CDRM name to the top of the default SSCP list, regardless of whether it is included in the list. Session setup requests are then sent to the SSCPs in the default SSCP list until the owning SSCP is found or the end of the list is reached.

If you coded NETID on the NETWORK definition statement, thereby associating it with the CDRSC, and the CDRM name is not the SSCP name of the CDRM in which the resource resides, the session setup could fail. See “VFYOWNER” on page 112 for more details.

If a NETID is not associated with the CDRSC, the CDRM operand, if coded, is used as an adjacent SSCP. Adjacent SSCP routing will be attempted first, before other entries in the adjacent SSCP table are tried.

Notes:

1. The preceding actions affect only the internal copy of the adjacent SSCP table maintained for each CDRSC. They do not affect the actual adjacent SSCP table. When the last session for this resource ends, the CDRM name is reset to the CDRM value specified on the CDRSC definition statement.
2. In the SSCP of the originating logical unit (OLU), the CDRM name of the destination logical unit (DLU), if identified, must be correct. If the CDRM name of the DLU is not correct, the operator must issue a MODIFY CDRM command to change the CDRM name.
3. You can permanently change the owning CDRM name with the MODIFY CDRM command. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY CDRM command.

cdrscname

```
►►──cdrscname──────────────────────────────────────────────────────────►◄
```

statements: *CDRSC*
dependencies: *none*

Specifies the name of a resource (a logical unit or application program) in another domain or network that this CDRSC is to represent.

cdrscname is required. *cdrscname* can be the name of the logical unit or application as known in this network (the alias name). If the resource is a VTAM application program, and the CDRSC is to satisfy the real name, *cdrscname* must be the same as the *name* (not the ACBNAME operand) of the application program's APPL definition statement in the VTAM definition in the other domain.

If you do not code the NETWORK statement, the *cdrscname* is either the real name of the resource or the alias name of the resource. If a CDRSC is predefined in the same domain for the real name, and another CDRSC is predefined in that domain for the alias name, the session setup fails. VTAM does not allow two static definitions of the same resource in a single domain.

In an SNA network interconnection, only one CDRSC identified by the name in *cdrscname* can be defined using NQNMODE=NAME (either by the start option or by the definition statement). Same-network CDRSCs and CDRSCs defined without a NETWORK statement are always considered to be defined using NQNMODE=NAME. If a CDRSC is defined using NQNMODE=NAME, then *cdrscname* cannot conflict with the name of any non-CDRSC resource in that SNA-interconnected network.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └─CONFGDS=name─────┘
```

statements: *VBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └─CONFGPW=password─┘
```

statements: *VBUILD*
dependencies: *CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you code CONFGDS.

CPNAME

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─CPNAME=control_point_name────┘
```

statements:	*GROUP, CDRSC*
dependencies:	*REGISTER=NO*

Identifies the resource's owning control point name. To allow definition of casually connected resources, this operand can be fully qualified with a network ID. Although it is not recommended, if you define an APPN resource to avoid APPN broadcasts to locate it, then CPNAME is required.

CPNAME can be the same name as the host's control point name (SSCP name). However, CPNAME cannot match the name of the CDRSC minor node.

If the CPNAME or NETSRVR name is qualified with a network ID, that network ID is used regardless of the NETID operand value. Otherwise, the value of the NETID operand is used to qualify the name. If NETID is not coded, CPNAME and NETSRVR are qualified with the host's network ID.

Note that for the cross-domain resource major node, CPNAME and REGISTER are conflicting options. If you code CPNAME and a REGISTER value of anything other than NO, an error message is issued and REGISTER is set to NO.

DLOGMOD

```
►►──┬──────────────────────────────────────┬──────────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name─┘
```

statements:	*GROUP, CDRSC*
dependencies:	*MODETAB*

Specifies the name of the default logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied default logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on logon mode entries, see "Logon Mode Table" on page 698.

EAS

```
   ┌─EAS=256──────────────┐
►►─┤                      ├──────────────────────────────────────────►◄
   └─EAS=number_of_sessions─┘
```

statements:	*GROUP, CDRSC*
dependencies:	*valid only for ILUs (LOCADDR=0)*
range:	*0–65535*

Specifies the estimated number of concurrent sessions that will be active with this independent logical unit.

If your estimated value is greater than 256, for better performance, use the next higher multiple of 256.

You can use this operand in conjunction with the BSBUF buffer pool start option to support peak session levels. See “Buffer Pool” on page 566 for information about BSBUF.

ENCR

```
   ┌─ENCR=OPT──────────┐
►►─┤                   ├──────────────────────────────────────────►◄
   │          ┌─OPT──┐ │
   └─ENCR=────┤      ├─┘
              ├─COND─┤
              ├─NONE─┤
              ├─REQD─┤
              └─SEL──┘
```

statements:	*GROUP, APPL*
dependencies:	*none*

Specifies whether this application program has any special requirements for enciphering and deciphering messages.

ENCR=COND
Specifies that this application program will establish an encrypted session if the session partner supports encryption and cryptographic product support is available. If support is not available at session start, a clear session (without encryption) is established.

VTAM starts sessions in the clear when:

- ENCR=COND is coded on the APPL definition of the PLU,
- The SLU has an APPL definition with ENCR=COND or an LU definition with ENCR=OPT
- A cryptograpic support is unavailable.

There is no attempt to start data encryption in the middle of a clear session when the encryption product becomes available again.

In VTAM, the COND option includes a test to determine if cryptograpic product support is available on the primary logical unit (PLU). However, if either end of a session specifies a higher level of encryption than ENCR=COND (ENCR=SEL or ENCR=REQD), an encrypted session is required and no test is made to determine if cryptographic product support is available.

ENCR=NONE
Specifies that this application program cannot have cryptographic sessions.

ENCR=OPT
Specifies that the application program has no special cryptographic requirements; its cryptographic capability is the same as the host processor's capability.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from the application program. If ENCR=REQD, this application program cannot be started unless the host with which it is associated can handle cryptographic sessions. ENCR=REQD also means that this application program cannot start a session with a logical unit that cannot encipher and decipher its messages.

ENCR=SEL
Specifies that this application program can choose which messages are enciphered by VTAM. ENCR=SEL also means that this application program cannot start a session with a logical unit that cannot encipher and decipher its messages. If you code ENCR=SEL, this application program cannot be started unless the host in which it resides can handle cryptographic sessions.

ENCRTYPE

```
                ┌─ENCRTYPE=DES──────┐
►►──────────────┤                   ├──────────────────────────────────────►◄
                └─ENCRTYPE=TDES24───┘
```

statements:	*GROUP, CDRSC*
dependencies:	*none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

Note: The ENCRTYPE value coded on the GROUP will be sifted down to the appropriate resource (CDRSC).

The SYSDEFed ENCRTYPE of a CDRSC resource can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to "MODEENT Macroinstruction" on page 699 for more information on MODEENT macroinstructions.

ISTATUS

```
►►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────────►◄
   │                         │
   └─ISTATUS=─┬─ACTIVE───┬───┘
              └─INACTIVE─┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies the initial status of this CDRSC.

ISTATUS=ACTIVE
Specifies that this resource is logically active to this domain, which means that session-establishment requests can be sent to the adjacent SSCP using the resource. The resource is not necessarily active in its own domain.

ISTATUS=INACTIVE
Specifies that this resource is not logically active to this domain, though it is not necessarily inactive in its own domain. Session requests from this domain to the resource will fail.

LOGAPPL

```
►►─┬──────────────────────────────┬──────────────────────────────────────►◄
   └─LOGAPPL=controlling_primary_lu─┘
```

statements: *GROUP, CDRSC*
dependencies: *NetView or TSO*

Identifies the name of the primary logical unit (PLU) that this logical unit will automatically log on to as a secondary logical unit (SLU) when this logical unit is activated. The name specified can also be a USERVAR, which will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

For independent logical units, the LOGAPPL operand establishes a session with the primary logical unit designated on the LOGAPPL operand if a session does not currently exist with this logical unit. The session is established regardless of whether sessions with other logical units exist. The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LUALIAS

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─LUALIAS=lualias──┘
```

statements:	*CDRSC*
dependencies:	*see description*

Defines a 1–8 character name used by application programs in this domain to represent a cross-network or cross-domain resource that is defined under a NETWORK definition statement that precedes the CDRSC definition statement.

Note: LUALIAS is valid only for cross-domain and cross-network resources that are defined under a NETWORK definition statement, and is recommended only for cross-network CDRSCs which are defined using NQNMODE=NQNAME because same-network CDRSCs and NQNMODE=NAME CDRSCs are known by their unqualified resource names.

Session requests from this domain's application programs are routed to the CDRSC, when the destination resource has the same name as defined by LUALIAS. Session requests from any resource other than a VTAM application in this domain are routed to the resource in this domain with the same name.

If you define two CDRSCs with the same LUALIAS name, only one can be active at a time. This means that a shadow resource with an LUALIAS name that duplicates a real resource with the same LUALIAS name, regardless of whether the resource names for the two resources are the same, cannot be activated until the real resource is deactivated. A shadow resource that attempts to become active and cannot because it has an LUALIAS name that duplicates an active resource is put into RESET state. After it is in RESET state, the major node of the shadow resource must be deactivated, the active resource with the LUALIAS name must be deactivated, and then the major node of the shadow resource must be activated before the shadow resource can be activated.

An LU in shadow (which does not have a duplicate LUALIAS name) also can be activated when it comes out of shadow if the LU's initial status (ISTATUS operand) was active while it was in shadow, and the CDRSC that held the LU in shadow was not inactivated.

Note: LUALIAS takes effect only in the following situations:

- When the LUALIAS name has been defined for a resource that is defined on the CDRSC definition statement that is preceded by a NETWORK definition statement.
- For session requests that cross the application program interface.
- For session requests whose origin or destination is the VTAM for which LUALIAS is coded. VTAMs along the session setup path are not involved with or affected by LUALIAS.

  Note: LUALIAS can be used in DISPLAY commands issued in the VTAM where LUALIAS is defined. LUALIAS cannot be used in MODIFY or VARY commands, and cannot be specified for LOGAPPL.

LUALIAS is optional for cross-network or cross-domain resources predefined with the NETWORK definition statement. If it is not defined and there are duplicate names in the SNA network interconnection, application programs can use the network-qualified name [if PARMS=(NQNAMES=YES) is specified on the ACB macroinstruction] or any of the other types of alias name translation, such as:
- Alias name translation with NetView
- Session management exit routine
- USERVAR

If you are predefining this resource just to specify an LUALIAS, consider coding LUAPFX on the APPL definiton statement which does not support network-qualified names and allow the CDRSC to be created dynamically. VTAM will dynamically gnerate an LUALIAS for each dynamic cross-network CDRSC as needed. See “LUAPFX” on page 48 for more information about dynamic LUALIAS.

MAXSESS

```
►►──┬──────────────────────────────────────────┬──────────────────────────────►◄
    └─MAXSESS=max_number_LU-LU_sessions────────┘
```

statements:	*GROUP, CDRSC*
dependencies:	*Independent LUs attached to an NCP*
range:	*1–65535*

Specifies the maximum number of concurrent LU-LU sessions in which an independent LU can participate. The MAXSESS value applies to each connection the LU is using to establish LU-LU sessions. For example, if MAXSESS=2 was specified for LUA and LUA appears on three PUs, LUA can have two sessions on each of the three PUs for a total of six sessions.

If you do not code MAXSESS, or code a value outside the range, the MAXSESS value set in the NCP definition is used.

MDLENT

```
►►──┬──────────────────────────────────────────┬──────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name───────┘
```

statements:	*GROUP, CDRSC*
dependencies:	*MDLTAB*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

If you omit the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application program in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MODETAB

```
    ┌─MODETAB=ISTINCLM──────────────┐
►►──┼───────────────────────────────┼──────────────────────────────────────►◄
    └─MODETAB=logon_mode_table_name─┘
```

statements: *GROUP, CDRSC*
dependencies: *none*

Specifies the name of a logon mode table to be used for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not code a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►──┬──────┬──────────────────────────────►◄  ►►──name──────────────────────────►◄
    └name──┘
```

statements:	VBUILD, NETWORK	*statements:*	GROUP

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On NETWORK: Identifies this statement in error messages from the VTAM system definition processor. To avoid confusion, use the same value for both *name* and the NETID operand. *name.* is optional, but its use is highly recommended.

On GROUP: Specifies the required name of the CDRSC group.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └NETID=network_id──┘
```

statements:	*NETWORK*
dependencies:	*none*

Specifies an optional 1–8 character identifier of the network in which the CDRSC, or set of CDRSCs, resides. This NETID pertains to the CDRSC, or set of CDRSCs, defined after the NETWORK definition statement and before any other NETWORK definition statement in the same set of definitions.

Notes:

1. If a CDRSC is defined with a network ID of the host network, the resource is considered real.
2. A CDRSC defined with a network ID of the host network must reside in the host network.
3. The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers (NETIDs). Although VTAM allows you to use #, other products might enforce this restriction.

NETSRVR

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └NETSRVR=network_node_server──┘
```

statements: *GROUP, CDRSC*
dependencies: *APPN only*

Identifies the resource's network node server. To allow predefinition of casually connected resources, this operand can be fully qualified with a network ID. If the operand is not network-qualified, the NETID value coded on the NETWORK statement is used.

The NETSRVR operand can have the same name as the host's control point name if the node is the network node server. However, NETSRVR cannot match the name of the CDRSC minor node.

Note: Specifying this operand does not initiate CP-CP sessions with the network node server. The name is used only to suggest a network node server for a control point.

NQNMODE

```
►►─┬─NQNMODE=value_of_nqnmode_start_option──────────┬──────────────────────────────────►◄
   │                                                │
   └─NQNMODE=─┬─value_of_nqnmode_start_option─┬─────┘
              ├─NAME──────────────────────────┤
              └─NQNMODE───────────────────────┘
```

statements: *GROUP, CDRSC*
dependencies: *valid only for resources defined under the NETWORK statement*

Indicates whether VTAM defines the cross-network resource by its non-network-qualified name and its network-qualified name or by only its network-qualified name.

NQNMODE is valid only for cross-domain and cross-network resources that are defined under a NETWORK definition statement. Same-network CDRSCs are always considered to be defined with NQNMODE=NAME because these resources are always identified by their non-network-qualified names.

When NQNMODE is specified on the CDRSC definition statement, it overrides the value specified on the NQNMODE start option. Dynamic CDRSCs are controlled by the NQNMODE start option.

Table 15 on page 109 shows the various ways NQNMODE applies to different types of CDRSCs.

NQNMODE=NAME
The cross-network resource is defined by both its network-qualified name and its non-network-qualified name. The same-network CDRSC is always considered to be defined using NQNMODE=NAME because it is always identified by its non-network-qualified name. This is the mode in which pre-V4 releases of VTAM operate.

If a CDRSC is defined using NQNMODE=NAME (either by definition or start option), then no other resource can be defined to that host by the same non-network-qualified name.

NQNMODE=NQNAME
The cross-network resources is defined by its network-qualified name and is not defined to this host by its non-network-qualified name.

If you have duplicate names in your SNA-interconnected networks and want to avoid using other forms of alias name translation, you can specify (either by definition or start option) NQNMODE=NQNAME.

Table 15. How NQNMODE Relates to CDRSCs

Type of CDRSC	CDRSC or Start Option	NQNMODE=NAME	NQNMODE=NQNAME
Predefined CDRSCs with NETID	CDRSC and start option (CDRSC overrides start option)	Cross-network resources defined by their non-network-qualified names and their network qualified names	Cross-network resources defined by their network-qualified names only
Predefined CDRSCs without NETID	Neither (NQNMODE not applicable)		
Dynamic CDRSCs defined by their network-qualified names and their implied alias names (consisting of their non-network-qualified names and the network identifiers of their session partners).	Start option	CDRSCs defined by their non-network-qualified names and their network qualified names	CDRSCs defined by their network-qualified names
Same-network CDRSCs and native independent LUs	Neither (NQNMODE not applicable)	Has no affect. Network-qualified name and non-network-qualified name are considered equivalent.	Has no affect. Network-qualified name and non-network-qualified name are considered equivalent.
Nonnatively attached independent LUs defined as LUs with LOCADDR=0 coded	Start option	CDRSCs defined by their non-network-qualified names and their network qualified names	CDRSCs defined by their network-qualified names only
Nonnatively attached independent LUs defined as CDRSCs	CDRSC and start option (CDRSC overrides start option)	CDRSCs defined by their non-network-qualified names and their network qualified names	CDRSCs defined by their network-qualified names only

PACING

```
►►─┬─PACING=1──────────────────┬──────────────────────────────────────────►◄
   │                           │
   └─PACING=─┬─1──────────────┬─┘
             └─pacing_number──┘
```

statements: *GROUP, CDRSC*
dependencies: *none*
range: *0–63*

Specifies how VTAM is to pace the flow of data from the boundary node, which performs pacing for a channel-attached SNA device, to the SLU. When the SLU and the PLU are in the same domain, the PACING value is ignored.

Notes:

1. PACING=0 indicates that no pacing is performed for session with the logical unit.
2. You can use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and use fixed pacing instead.

   To use the FIXED option, you must code (N,,FIXED). The second comma represents M or a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

REGISTER

```
►►─┬─REGISTER=NO──────────────┬──────────────────────────────────────────────►◄
   │                          │
   └─REGISTER=─┬─NO──────────┬─┘
               │             │
               ├─CDSERVR─────┤
               │             │
               └─NETSRVR─────┘
```

statements:	*GROUP, CDRSC*
dependencies:	*CPNAME; APPN only*

Specifies how a resource should be registered on a definition statement.

Note that for the cross-domain resource major node, CPNAME and REGISTER are conflicting options. If you code CPNAME and a REGISTER value of anything other than NO, an error message is issued and REGISTER is set to NO.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to the network node server, but that directory resource registration should not be requested for it.

REGISTER=NO
Indicates that the resource should not be registered.

RESSCB

```
►►─┬─RESSCB=0──────────────────────────────┬──────────────────────────────────►◄
   │                                       │
   └─RESSCB=number_of_session_control_blocks─┘
```

statements:	*GROUP, CDRSC*
dependencies:	*valid only for type 2.1*
range:	*0–65535*

Specifies the number of session control blocks reserved by the NCP for the logical unit.

This operand applies only to type 2.1 peripheral nodes.

For information about NCP restrictions on this operand, refer to the *NCP, SSP, and EP Resource Definition Reference*.

SPAN

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └SPAN=span_name────────┘
```

statements:	*GROUP, CDRSC*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView Program checks the SPAN value, but VTAM ignores it.

SRCOUNT

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └SRCOUNT=number_of_search_requests┘
```

statements:	*GROUP, CDRSC*
range:	*0–65535*
dependencies:	*SRCHRED start option*

Specifies for this resource how many search requests can be limited before VTAM performs another resource discovery search. It is meaningful only when search reduction is active (SRCHRED=ON). This operand overrides the value of the SRCOUNT start option for this CDRSC.

Notes:

1. If SRCOUNT=0 is coded, then the search count threshold is not used and the SRTIMER value alone will control when to attempt another search.
2. If neither SRCOUNT or SRTIMER is coded on the CDRSC statement and SRCHRED=ON, then the SRCOUNT and SRTIMER start option values will be used.
3. If SRCOUNT is coded, an APPN directory entry will be created (if one does not exist for the resource) in addition to the CDRSC defined by this definition statement.

SRTIMER

```
►►──┬──────────────────────────┬──────────────────────────────────────►◄
    └SRTIMER=number_of_seconds─┘
```

statements:	*GROUP, CDRSC*
range:	*0–65535*
dependencies:	*SRCHRED start option*

Specifies the number of seconds that a search reduction entry limits incoming search requests or session requests. SRTIMER is meaningful only when search reduction is active (SRCHRED=ON). This operand overrides the value of the SRTIMER start option for this CDRSC.

Notes:

1. If SRTIMER=0 is coded, then the search time threshold is not used and the SRCOUNT value alone will control when to attempt another search.
2. If neither SRTIMER or SRCOUNT is coded on the CDRSC statement and the SRCHRED=ON, then the SRCOUNT and SRTIMER start option values will be used.
3. If SRCOUNT is coded, an APPN directory entry will be created (if one does not exist for the resource) in addition to the CDRSC defined by this definition statement.

TYPE

```
►►──TYPE=CDRSC──────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Indicates a CDRSC major node. This operand is required.

VFYOWNER

```
                ┌─VFYOWNER=NO──────────┐
►►──────────────┼──────────────────────┼──────────────────────────────►◄
                │              ┌─NO──┐ │
                └─VFYOWNER=────┼─────┼─┘
                               └─YES─┘
```

statements:	*GROUP, CDRSC*
dependencies:	*subarea sessions*

Specifies whether owner verification takes place when sessions are attempted with this resource.

Note: Owner verification is supported only in subarea sessions.

VFYOWNER=YES
Specifies that the session is rejected if the owner specified on the CDRSC does not match the owner specified in the session initiation request.
VFYOWNER=YES is valid only if:

- The CDRSC is predefined with the NETID operand on the NETWORK definition statement.
- You code CDRM=*cdrmname*.

VFYOWNER=NO
Specifies that, if the owning SSCP received on an initiation request does not match the owner specified on the CDRSC, the CDRSC is updated with the owning SSCP name received, unless the CDRSC is across the APPN-subarea boundary and CDRM=*cdrmname* is coded.

VPACING

```
►►─┬─VPACING=2──────────────────────┬──────────────────────────────────────►◄
   │                                │
   └─VPACING=─┬─2──────────────────┬┘
              └─number_of_requests─┘
```

statements:	*GROUP, CDRSC*
dependencies:	*none*
range:	*0–63*

Specifies how VTAM paces the flow of data from the boundary node containing the PLU to the boundary node that performs pacing for a channel-attached SNA device.

number_of_requests specifies the maximum number of normal-flow requests that the PLU sends for a given LU-LU session before waiting for a pacing response. No further normal-flow requests can be sent to the logical unit until it is ready to receive them.

Note: You can use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and use fixed pacing instead.

To use the FIXED option, you must code (N,,FIXED). The second comma represents M or a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Cross-Domain Resource Manager (CDRM) Major Node

To define a cross-domain resource manager (CDRM) major node, code a VBUILD definition statement for the major node, and a CDRM definition statement for each minor node.

You can define one or more major nodes for CDRMs.

For SNA network interconnection, define a cross-domain resource manager major node with:
- A VBUILD definition statement
- Optional NETWORK definition statements
- One or more CDRM definition statements
- Optional GWPATH definition statements

Note: Within a domain, one cross-domain resource manager, called an external CDRM, must be defined for every other domain (in this network or another network) with which this domain directly communicates. Each domain must also have only one definition for its own cross-domain resource manager, called the host CDRM.

Table 16. Cross-Domain Resource Manager Major Node Definition Statement and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CDRM	131	R			

Table 16. Cross-Domain Resource Manager Major Node Definition Statement and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		CONFGDS	122				
		CONFGPW	122				CONFGDS
name	NETWORK	NETID					
cdrmname	CDRM	SUBAREA	130	R			
		CAPACITY	120			8K	VRTG=YES; APPN only
		CDRDYN	120			NO	
		CDRSC	121			REQ	SSCP-SSCP sessions
		COSTBYTE	122			0	VRTG=YES; APPN only
		COSTTIME	123			0	VRTG=YES; APPN only
		DISJOINT	123			NO	APPN only
		ELEMENT	123			1	
		GWSELECT	125			YES	cross-net CDRMs
		HPR	125			YES	VRRTG=YES; APPN only; NODETYPE; HOSTSA;external CDRM definitions only
		ISTATUS	125			ACTIVE	
		NN	127				VRTG=YES; APPN only
		PDELAY	128			TERRESTR	VRTG=YES; APPN only
		RECOVERY	128			YES	
		SECURITY	129			UNSECURE	VRTG=YES; APPN only
		SPAN	129				NetView
		TGP	130				VRTG=YES; APPN only
		UPARM1	131			128	VRTG=YES; APPN only
		UPARM2	131			128	VRTG=YES; APPN only
		UPARM3	131			128	VRTG=YES; APPN only
		VPACING	132			63	
		VRTG	132			VRTG start option	APPN only; NODETYPE; HOSTSA
		VRTGCPCP	133			VRTGCPCP start option	VRTG=YES; APPN only; NODETYPE; HOSTSA
name	GWPATH	ADJNET	118			NETWORK	
		ADJNETCS	119			ISTVTCOS	
		ADJNETEL	119			1	ADJNET

Table 16. Cross-Domain Resource Manager Major Node Definition Statement and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		ADJNETSA	119				
		ELEMENT	123			1	
		GWN	124				PUNAME on BUILD
		SUBAREA	130				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement to define a cross-domain resource manager major node.

NETWORK Definition Statement

Code the NETWORK definition statement in an SNA-interconnected network, to name the network in which each of the CDRMs being defined resides. The NETWORK definition statement appears after the VBUILD definition statement and immediately before one or more CDRM definition statements defining CDRMs residing in the network.

You can include more than one NETWORK definition statement in a CDRM major node definition. If you include consecutive NETWORK definition statements (without intervening CDRM definitions), only the last one is used.

If no NETWORK definition statement exists before one or more CDRM definition statements, the CDRMs are assumed to reside within this VTAM network. The network name for this network is the name specified on the NETID start option for the VTAM in which the definition is filed.

Note: There are two considerations regarding a receiving (internal) CDRM and session setup:

- If the receiving CDRM is not a gateway SSCP, and its definition for an external CDRM does not include NETID, then its VTAM will accept whatever NETID is sent during session setup, regardless of whether it is correct. The CDRM session will set up, but LU-LU sessions will not. To prevent the above, code GWSSCP=YES on the receiving CDRM definition, or code NETID on the NETWORK definition statement of all external CDRMs.
- If the receiving CDRM is a gateway SSCP, and no NETID is coded for an external CDRM, then the NETID sent during session setup must match the value on the NETID start option of the receiving host or the session will not set up.

CDRM Definition Statement

Code a CDRM definition statement to define the host CDRM and each external CDRM that will be in direct communication with this domain. Define the host CDRM before defining any external CDRMs.

GWPATH Definition Statement

Code a GWPATH definition statement to define a possible cross-network session path between the gateway host CDRM and a CDRM in another network.

The gateway path (GWPATH) definition statement is optional. You should code a GWPATH definition statement only if VTAM is started with GWSSCP=YES.

One or more GWPATH definition statements can follow a CDRM definition statement for a CDRM in another network. VTAM processes GWPATH definition statements in the order in which they are assigned. Before VTAM uses the list, the GWPATH definition statement assigned for the SSCP-SSCP session is placed first in the list.

Note: The operands on the GWPATH definition statement are listed here in alphabetical order for easy reference. However, you might find it easier to study and code them in this order: GWN, SUBAREA, ELEMENT, ADJNET, ADJNETSA, ADJNETEL, and ADJNETCS.

Full Syntax

The full syntax for the cross-domain manager (CDRM) major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=CDRM──────────────────────────────────►
    └name──┘

►──┬──────────────────────────────────────────────────────────┬──►◄
   ├─CONFGDS=confgds_name──────────────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────────────┘
```

NETWORK

```
►►──┬──────┬──NETWORK──┬──────────────────────┬──────────────────►◄
    └name──┘           └─NETID=network_id─────┘
```

CDRM

```
                                   ┌─CAPACITY=8K──────────────┐
►►──cdrmname──CDRM──SUBAREA=suba_addr──┤                          ├──►
                                   └─CAPACITY=number_of_bits──┘
```

Cross-Domain Resource Manager Major Node

```
►─┬─CDRDYN=NO──────────┬─┬─CDRSC=REQ──────────┬─┬─COSTBYTE=0──────────────┬──►
  │                    │ │                    │ │                         │
  └─CDRDYN=─┬─NO──┬────┘ └─CDRSC=─┬─REQ─┬──────┘ └─COSTBYTE=cost_per_byte──┘
            └─YES─┘               └─OPT─┘

►─┬─COSTTIME=0──────────────┬─┬─DISJOINT=NO──────────┬──────────────────────►
  │                         │ │                      │
  └─COSTTIME=cost_per_unit_time─┘ └─DISJOINT=─┬─NO──┬──┘
                                              └─YES─┘

►─┬─ELEMENT=1──────────────┬─┬─GWSELECT=YES──────────┬─┬─HPR=YES──────────┬──►
  │                        │ │                       │ │                  │
  └─ELEMENT=element_address─┘ └─GWSELECT=─┬─YES─┬────┘ └─HPR=─┬─NO──┬────┘
                                          └─NO──┘              └─YES─┘

                                                   ┌─PDELAY=TERRESTR──────┐
                                                   │                (1)   │
►─┬─ISTATUS=ACTIVE──────────┬─┬──────────────────┼─PDELAY=NEGLIGIB──────┤──►
  │                         │ │                  │                      │
  └─ISTATUS=─┬─ACTIVE──┬────┘ └─NN=─┬─NO──┬──────┘ └─PDELAY=─┬─TERRESTR─┬─┘
             └─INACTIVE─┘            └─YES─┘                  ├─LONG────┤
                                                              ├─NEGLIGIB─┤
                                                              └─PACKET──┘

►─┬─RECOVERY=YES────────────┬─┬─SECURITY=UNSECURE──────────┬─┬──────────────┬──►
  │                         │ │                            │ │              │
  └─RECOVERY=─┬─YES─┬─────────┘ └─SECURITY=─┬─UNSECURE─┬────┘ └─SPAN=span_name─┘
              └─NO──┘                      ├─PUBLIC───┤
                                           ├─UNDERGRO─┤
                                           ├─SECURE───┤
                                           ├─GUARDED──┤
                                           ├─ENCRYPT──┤
                                           └─SHIELDED─┘

►──┬──────────────────┬─┬─UPARM1=128──────────────────────┬──────────────────►
   │                  │ │                                 │
   └─TGP=tgp_name─────┘ └─UPARM1=user-defined_operand_value─┘

►─┬─UPARM2=128──────────────────────┬─┬─UPARM3=128──────────────────────┬──►
  │                                 │ │                                 │
  └─UPARM2=user-defined_operand_value─┘ └─UPARM3=user-defined_operand_value─┘
```

Cross-Domain Resource Manager Major Node

```
►─┬─VPACING=63──────────────────┬─┬─VRTG=value_of_vrtg_start_option─┬──────────────►
  └─VPACING=number_of_requests─┘ └─VRTG=──┬─YES─┬──────────────────┘
                                          └─NO──┘

►─┬─VRTGCPCP=value_of_vrtgcpcp_start_option─┬──────────────────────────────────►◄
  └─VRTGCPCP=──┬─YES─┬──────────────────────┘
               └─NO──┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

GWPATH

```
►►──┬──────┬──GWPATH──┬─ADJNET=netid_of_preceding_network_def_statement─┬──────────►
    └─name─┘          └─ADJNET=adjacent_network_identifier──────────────┘

►─┬─ADJNETCS=ISTVTCOS──────────────┬──────────────────────────────────────────►
  └─ADJNETCS=adjacent_network_cos_name─┘

►─┬─ADJNETEL=1──────────────────────┬─┬──────────────────────────────────────►
  └─ADJNETEL=adjacent_network_element─┘ └─ADJNETSA=adjacent_network_subarea─┘

►─┬─ELEMENT=1──────────────┬─┬──────────────────┬─┬──────────────────┬──────►◄
  └─ELEMENT=element_address─┘ └─GWN=gateway_ncp_name─┘ └─SUBAREA=suba_addr─┘
```

Operand Descriptions

The following section describes the operands you can code for the cross-domain resource manager (CDRM) major node.

ADJNET

```
►►─┬─ADJNET=netid_of_preceding_network_def_statement─┬──────────────────────────►◄
   └─ADJNET=adjacent_network_identifier──────────────┘
```

statements:	*GWPATH*
dependencies:	*none*

Specifies, for this cross-network session path to the CDRM through the associated gateway NCP, the ID of the network that is adjacent to the gateway SSCP's network. The gateway NCP is determined either by the GWN operand or by the SUBAREA operand.

If you do not code ADJNET, the adjacent network defaults to the ID of the network defined on the preceding NETWORK definition statement. The value coded must not be the same as the host's network ID.

Note: In a back-to-back configuration, if multiple GWPATH definition statements are filed describing multiple GWN paths to the same CDRM, and those statements specify the same ADJNET, the ADJNETSA and ADJNETEL values must agree on all the GWPATH definition statements.

ADJNETCS

```
►►─┬─ADJNETCS=ISTVTCOS──────────────────────┬──────────────────────────────────►◄
   └─ADJNETCS=adjacent_network_cos_name─────┘
```

statements:	*GWPATH*
dependencies:	*none*

Specifies the 8-character name that defines the class-of-service (COS) table entry to be used for the SSCP-SSCP session. The table entry is located in the network that is adjacent to the gateway SSCP's network.

This operand must be coded if you do not want VTAM to go to the NetView alias application to determine the COS name for the SSCP-SSCP session. If this operand is not coded, the adjacent COS name defaults to ISTVTCOS, and the alias application, if active, is called with the default name.

ADJNETEL

```
►►─┬─ADJNETEL=1──────────────────────────┬──────────────────────────────────►◄
   └─ADJNETEL=adjacent_network_element───┘
```

statements:	*GWPATH*
dependencies:	*ADJNET*
range:	*1–32767*

Specifies the element address of the CDRM for the adjacent network. Code ADJNETEL if the actual network of the CDRM is not the same as ADJNET and the default value of 1 is not correct.

Although it is not required, you should code ADJNETEL for all GWPATH definition statements. Its value should be the same as that defined on the ELEMENT operand of the GWNAU definition statement. If the GWNAU definition statement is not coded, ADJNETEL should be the same as the element number of the adjacent SSCP.

ADJNETSA

```
►►─┬──────────────────────────────────────┬──────────────────────────────────►◄
   └─ADJNETSA=adjacent_network_subarea────┘
```

statements:	*GWPATH*
dependencies:	*none*
range:	*1–65535*

Specifies the subarea address of the CDRM for the adjacent network. Code ADJNETSA for all GWPATH definition statements.

CAPACITY

```
►►─┬─CAPACITY=8K──────────────────┬──────────────────────────────────────►◄
   │                              │
   └─CAPACITY=number_of_bits──────┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*
range:	*1Kb—1000Mb*

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either Kb per second (for example, 100K) or Mb per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Note: Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CDRDYN

```
►►─┬─CDRDYN=NO────────────────────┬──────────────────────────────────────►◄
   │                              │
   │              ┌─NO──┐         │
   └─CDRDYN=──────┴─YES─┴─────────┘
```

statements:	*CDRM*
dependencies:	*none*

Specifies whether the host CDRM is authorized to dynamically define CDRSC representations of cross-domain or cross-network resources when a session request is received from or sent to an adjacent SSCP or CP.

CDRDYN=NO
Causes the failure of any cross-domain or cross-network session request for a cross-domain or cross-network resource that has no predefined CDRSC entry.

CDRDYN=YES
Authorizes dynamic definition of cross-domain or cross-network resources by the host CDRM, eliminating the need to predefine certain CDRSCs. Both origin logical units and destination logical units can be dynamically defined. Note that for subarea nodes (HOSTSA start option is specified), the DYNLU=YES operand is meaningful only when CDRDYN=YES.

Notes:

1. To allow dynamic resource definition, code CDRDYN=YES on the CDRM definition statement for the VTAM host CDRM and CDRSC=OPT on the CDRM definition statement for the owning SSCP.
2. An intermediate host can also build dynamic definitions for CDRSCs.
3. The CDRDYN start option overrides the CDRDYN operand value. You can use the MODIFY VTAMOPTS command to change the value of CDRDYN.

cdrmname

```
►►──cdrmname──────────────────────────────────────────────────────────►◄
```

statements:	*CDRM*
dependencies:	*none*

Specifies, in a multiple-domain network, the 1–8 character name of the cross-domain resource manager minor node of this network. For CDRMs in other networks, *cdrmname* is the name by which this VTAM identifies the other-network CDRMs. *cdrmname* is required for this network and other networks.

Because *cdrmname* is used in the ID operand of VTAM operator commands, it must be a name unique to this VTAM. As a result, the name fields of all CDRM definition statements that are active at one time must be unique and not conflict with other resource names in the same network. Because the CDRM definition statement in an adjacent SSCP table requires the SSCPNAME of an external CDRM when the CDRSC associated with the table has been coded with a NETWORK definition statement, you might have to change the labels on the CDRM definition statements.

To simplify system definition and the network operator's job, use the same name for the host CDRM and for the SSCPNAME start option if the NODETYPE start option is specified. If NODETYPE is coded in the start list and the names do not match, the following actions are taken:

- The host CDRM name is ignored and the SSCPNAME is required.
- Messages are generated to inform the operator that the SSCPNAME is being used instead of the host CDRM name.

Note: CDRM sessions might be affected by the name change. If you do not use the same *cdrmname*, the session might fail in some cross-network configurations.

CDRSC

```
►►──┬─CDRSC=REQ──────────┬──────────────────────────────────────────►◄
    │                    │
    └─CDRSC=──┬─REQ─┬────┘
              └─OPT─┘
```

statements:	*CDRM*
dependencies:	*SSCP-SSCP sessions*

Determines whether resources owned by the CDRM being defined by this statement can be dynamically defined at the time a session request is received from or sent to the external CDRM that manages the resource. The value you specify determines whether dynamic CDRSC definitions are allowed when cross-domain or cross-network session requests are received from or sent to certain domains.

Note: CDRSC is used only for SSCP-SSCP sessions.

CDRSC=OPT
Authorizes dynamic definition of cross-domain or cross-network resources from this CDRM, eliminating the need to predefine certain CDRSCs.

To allow dynamic resource definition, code CDRSC=OPT on the CDRM definition statement for the owning SSCP and code CDRDYN=YES on the CDRM definition statement for the VTAM host CDRM.

CDRSC=REQ
Causes the failure of any cross-domain or cross-network request to or from this CDRM for a cross-domain resource that has no predefined CDRSC entry.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─CONFGDS=name─────┘
```

statements: *VBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─CONFGPW=password─┘
```

statements: *VBUILD*
dependencies: *CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you code CONFGDS.

COSTBYTE

```
    ┌─COSTBYTE=0──────────────┐
►►──┼─────────────────────────┼──────────────────────────────────►◄
    └─COSTBYTE=cost_per_byte──┘
```

statements: *CDRM*
dependencies: *VRTG=YES coded; APPN only*
range: *0–255*

Specifies a cost-per-byte-transmitted to be associated with the transmission group. A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
►►─┬─COSTTIME=0──────────────────────┬──────────────────────────────────────►◄
   │                                 │
   └─COSTTIME=cost_per_unit_time─────┘
```

statements: *CDRM*
dependencies: *VRTG=YES coded; APPN only*
range: *0–255*

Specifies a cost-per-unit-of-time to be associated with the transmission group. A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

DISJOINT

```
►►─┬─DISJOINT=NO──────────────┬──────────────────────────────────────────►◄
   │                          │
   │              ┌─NO─┐      │
   └─DISJOINT=────┴────┴──────┘
                  └─YES─┘
```

statements: *CDRM*
dependencies: *APPN only*

Indicates whether a search from an SSCP should be sent to another SSCP, when the SSCPs have matching NETIDs and their only connection is through an APPN network. If the NETIDs match, the coding of the DISJOINT operand determines whether the search is sent.

In effect, coding DISJOINT=NO restricts APPN searches, especially broadcast searches, from entering the subarea network multiple times through different interchange nodes (or even through the same interchange node.)

DISJOINT=NO
Indicates that the APPN network should not send the search to the second SSCP.

DISJOINT=YES
Indicates that the APPN network should send the search to the second SSCP.

Note: If the NETIDs of the SSCPs do not match, the search is always sent to the second SSCP, regardless of the coding of the DISJOINT operand.

ELEMENT

```
   ┌─ELEMENT=1──────────────┐
►►─┤                        ├──────────────────────────────────────────►◄
   └─ELEMENT=element_address┘
```

statements:	*CDRM, GWPATH*
dependencies:	*none*
range:	*0–32767*

On CDRM:

Specifies the element part of the network address of the CDRM. This operand, with the SUBAREA value, defines the network address of the CDRM.

For the host CDRM, the ELEMENT value must be 1. For external CDRMs (CDRMs in other domains or networks), this number can be a decimal integer in the range 0–32767. The default value is 1 for both same- and cross-network CDRMs.

ELEMENT=0 specifies a TCAM CDRM.

On GWPATH:

Specifies the element part of the alias address for the CDRM in the VTAM network. The alias address is an address in the associated gateway NCP. The gateway SSCP uses this value only if it does not have a session with the associated gateway NCP when it starts the CDRM for the other network.

element address must be a decimal integer in the range 0–32767. If you use the default of 1, or if you code ELEMENT and SUBAREA, VTAM assigns the network address to the CDRM if it is active. The network address must be unique in the network. The element address must match the element address predefined in the GWNAU definition statement for the CDRM.

GWN

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─GWN=gateway_ncp_name─┘
```

statements:	*GWPATH*
dependencies:	*PUNAME on BUILD statement*

Provides the 1–8 character name of the gateway NCP that accesses the CDRM. If you code PUNAME in the NCP BUILD definition statement, *gateway_ncp_name* must match the PUNAME. If you do not code PUNAME, *gateway_ncp_name* is the name used to file the gateway NCP definition statements.

If you do not code this operand, you must code the SUBAREA operand, because VTAM uses the SUBAREA operand value to determine which gateway NCP it uses to access the CDRM. It is to your advantage to omit GWN if you use different node names, but the same subarea number, for a gateway NCP. (For example, the NCP might have one name in a test system and another name in a production system.) In that case, VTAM uses the subarea number when it accesses the CDRM, without regard to the node name.

GWSELECT

```
►►─┬─GWSELECT=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─GWSELECT=─┬─YES─┬─────┘
               └─NO──┘
```

statements: *CDRM*
dependencies: *cross-network CDRMs*

Defines whether the host CDRM is to perform gateway path selection for LU-LU sessions flowing to or through this cross-network external CDRM.

GWSELECT=NO
Prevents the host CDRM from performing gateway path selection for an LU-LU session flowing to or through this cross-network external CDRM.

GWSELECT=YES
Authorizes the host CDRM to perform gateway path selection for an LU-LU session flowing to or through this cross-network external CDRM.

Note: Code GWSELECT only for cross-network CDRMs. It is ignored for all others.

HPR

```
►►─┬─HPR=YES──────────┬──────────────────────────────────────────►◄
   │                  │
   └─HPR=─┬─NO──┬────┘
          └─YES─┘
```

statements: *CDRM*
dependencies: *VRTG=YES; APPN only; NODETYPE; HOSTSA; external CDRM definitions only*

Specifies whether high performance routing (HPR) support is enabled over the VR-based TG to a CDRM. This operand is applicable only to external CDRM definitions; it is not applicable to host CDRM definitions.

HPR=YES
Indicates that HPR support is provided over the VR-based TG to this CDRM.

HPR=NO
Indicates that HPR support is not provided over the VR-based TG to this CDRM.

Note: HPR defaults to NO when the HPR start option is specified as (RTP,NONE) or (ANR,NONE). If the HPR start option is specified as NONE, this operand is ignored.

ISTATUS

Cross-Domain Resource Manager Major Node

```
►►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────►◄
   │                         │
   │            ┌─ACTIVE───┐  │
   └─ISTATUS=───┴──────────┴──┘
                └─INACTIVE─┘
```

statements: *CDRM*
dependencies: *none*

Specifies the initial status of the host or external CDRM.

For the host CDRM:

ISTATUS=ACTIVE
Specifies that it can establish a session with an external CDRM.

ISTATUS=INACTIVE
Specifies that it cannot establish a session with an external CDRM until it is started.

For the external CDRM:

ISTATUS=ACTIVE
Specifies that a session is attempted between the host in which this statement is being defined and the CDRM named by this definition statement. Any attempt to establish this session fails if either the host CDRM or the external CDRM is not active. Code ISTATUS=ACTIVE only if:

- This CDRM is in the same major node as the host CDRM (for which ISTATUS=ACTIVE is coded), or
- This CDRM major node is to be started after the host CDRM is active.

ISTATUS=INACTIVE
Specifies that a session between CDRMs is not automatically attempted when the CDRM major node is started.

After its major node has been started, a CDRM defined as INACTIVE can be started with the VARY ACT command, or for a CDRM in another domain, by receiving a session-activation request from the domain.

name

```
►►─┬──────┬──────────────────────────────────────────────────────────►◄
   └─name─┘
```

statements: *VBUILD, NETWORK, GWPATH*
dependencies: *none*

On VBUILD: Specifies the optional name of this major node.

If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the

VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On NETWORK: Identifies this NETWORK definition statement in error messages from the VTAM system definition processor. To avoid confusion, you should code the same value for both *name* and the NETID operand. *name* is optional, but its use is highly recommended.

On GWPATH: Identifies this GWPATH definition statement in error messages from the VTAM system definition processor. *name* is optional, but its use is highly recommended.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └NETID=network_id──┘
```

statements:	*NETWORK*
dependencies:	*none*

Specifies a 1–8 character identifier for the network in which the CDRM or set of CDRMs reside. This NETID pertains to the CDRM or set of CDRMs defined after the NETWORK definition statement and before any other NETWORK definition statement in the same set of definitions.

NETID is required only on external CDRMs if GWSSCP=YES; otherwise, NETID is optional.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

NN

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └NN=──┬─NO──┬──────┘
          └─YES─┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*

Specifies that the virtual-route-based transmission group represents a connection to an interchange node or a migration data host. When this parameter is specified, the network capabilities specified by the adjacent CP during VR-based transmission group activation are validated against the coded value. If the indicators do not match, both the VR-based transmission group and SSCP-SSCP session will be terminated.

When this operand is not specified, no network capabilities are validated.

NN=YES
Indicates that the adjacent node must be an interchange network node.

NN=NO
Indicates that the adjacent node must be a migration data host.

PDELAY

```
  ┌─PDELAY=TERRESTR──────┐
  │                (1)   │
  ├─PDELAY=NEGLIGIB──────┤
►►┼──────────────────────┼──────────────────────────────────────────►◄
  │          ┌─TERRESTR─┐│
  └─PDELAY=──┼──────────┼┘
             ├─LONG─────┤
             ├─NEGLIGIB─┤
             └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

RECOVERY

```
  ┌─RECOVERY=YES─────┐
►►┼──────────────────┼──────────────────────────────────────────►◄
  │          ┌─YES─┐ │
  └─RECOVERY=┼─────┼┘
             └─NO──┘
```

statements:	*CDRM*
dependencies:	*none*

Determines whether the SSCP-SSCP session between the named CDRM and the host CDRM should be restarted automatically when a session fails.

Both the host CDRM and the external CDRM must have RECOVERY=YES to enable automatic recovery.

RECOVERY=NO
Specifies that the SSCP-SSCP session should not be restarted automatically.

RECOVERY=YES
Specifies that the SSCP-SSCP session should be restarted automatically.

SECURITY

```
  ┌─SECURITY=UNSECURE──────┐
►►┼────────────────────────┼──────────────────────────────────────────────►◄
  │                        │
  │              ┌─UNSECURE─┐
  └─SECURITY=────┼──────────┼
                 ├─PUBLIC───┤
                 ├─UNDERGRO─┤
                 ├─SECURE───┤
                 ├─GUARDED──┤
                 ├─ENCRYPT──┤
                 └─SHIELDED─┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SPAN

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └─SPAN=span_name───┘
```

statements:	*CDRM*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SUBAREA

On CDRM

```
►►──SUBAREA=suba_addr──────────────────────►◄
```

statements:	CDRM
dependency:	none
range:	1–value of MXSUBNUM start option

On GWPATH

```
►►──┬──────────────────────┬──────────────►◄
    └─SUBAREA=suba_addr────┘
```

statements:	GWPATH
dependency:	none
range:	1–value of MXSUBNUM start option

On CDRM: Specifies the number of the subarea in which this CDRM resides.

The SUBAREA value for each subarea used in cross-domain or cross-network communications must be unique throughout the network, but not throughout a set of SNA interconnected networks. For a host CDRM, this SUBAREA value must be the same as the host's subarea specified by the HOSTSA start option.

If no GWPATH definition statements follow the other-network CDRM definition statement, code the SUBAREA operand on that CDRM definition statement. The SUBAREA operand must represent a valid subarea number in this network for a gateway NCP.

GWPATH provides subarea information. It is not required for cross-domain communication, but is required for cross-network communication if the following conditions exist:

- CDRM is defined for the same network
- GWSSCP=YES in the START options and this CDRM is defined for cross-network sessions

On GWPATH: Specifies the subarea part of the real address for the CDRM in the network of the gateway SSCP. The real address is an address in the associated gateway NCP.

For a gateway SSCP, code SUBAREA if you do not code GWN. You can use SUBAREA instead of GWN when the gateway NCP has different node names but always has the same subarea number. Refer to the description of the GWN operand.

The gateway SSCP uses this value if you do not code GWN, or if it does not have a session with the associated gateway NCP when VTAM starts the other network's CDRM.

TGP

```
►►──┬──────────────────────┬──────────────────────────────────────►◄
    └─TGP=tgp_name─────────┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*

Specifies the name of a transmission group (TG) profile definition. The characteristics defined in the TG profile (along with any modifiers specified on this statement) become the characteristics of the VR-based transmission group. If *tgp_name* is not specified or the specified name cannot be found, the default TG characteristics are assigned.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TYPE

```
►►──TYPE=CDRM──────────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Indicates a CDRM major node. This operand is required.

UPARM1

```
►►─┬─UPARM1=128──────────────────────────┬──────────────────────────────►◄
   └─UPARM1=user-defined_operand_value──┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*
range:	*0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
►►─┬─UPARM2=128──────────────────────────┬──────────────────────────────►◄
   └─UPARM2=user-defined_operand_value──┘
```

statements:	*CDRM*
dependencies:	*VRTG=YES coded; APPN only*
range:	*0–255*

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────►◄
   └─UPARM3=user-defined_operand_value──┘
```

statements: *CDRM*
dependencies: *VRTG=YES coded; APPN only*
range: *0–255*

Specifies a user-defined operand value.

VPACING

```
►►─┬─VPACING=63──────────────────┬──────────────────────────────────────────►◄
   │                             │
   └─VPACING=number_of_requests──┘
```

statements: *CDRM*
dependencies: *none*
range: *0–63*

Specifies the maximum number of requests that other CDRMs can send to this CDRM before waiting to receive a pacing response. This value is exchanged when a session is established between CDRMs. This operand is effective only when the CDRM being defined is the host's own CDRM. VPACING affects only SSCP-SSCP session traffic.

If you specify a value greater than 63, the maximum value of 63 is used. If VPACING=0, no pacing is done.

The value of VPACING can be expressed with leading zeros and is limited to eight digits.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

VRTG

```
►►─┬─VRTG=value_of_vrtg_start_option─┬──────────────────────────────────────►◄
   │                                 │
   └─VRTG=──┬─YES─┬──────────────────┘
            └─NO──┘
```

statements: *CDRM*
dependencies: *NODETYPE and HOSTSA start options coded; APPN only*

Indicates for this node, whether virtual-route-based transmission group connections are requested when SSCP-SSCP sessions are established. This operand is valid only for interchange nodes and migration data hosts (when start options NODETYPE and HOSTSA are specified).

NO
Indicates that VR-based transmission group connection is not requested when SSCP-SSCP sessions are established.

YES
Indicates that VR-based transmission group connection is requested when SSCP-SSCP sessions are established.

VRTGCPCP

```
   ┌─VRTGCPCP=value_of_vrtgcpcp_start_option─┐
►►─┤                                         ├──────────────────────────────►◄
   └─VRTGCPCP=──┬─YES─┬──────────────────────┘
                └─NO──┘
```

statements:	*CDRM*
dependencies	*VRTG=YES coded; APPN only NODETYPE and HOSTSA start options coded*

Indicates whether CP-CP sessions are supported over the virtual-route-based transmission group. VRTGCPCP is valid only for interchange nodes and migration data hosts (when start options NODETYPE and HOSTSA are specified).

NO
Indicates that CP-CP sessions are not supported over the VR-based transmission group.

YES
Indicates that CP-CP sessions are supported over the VR-based transmission group.

External Communication Adapter (XCA) Major Node

The external communication adapter (XCA) major node defines VTAM's connections to the following shared access transport facilities (SATFs):

- Local area networks (LANs)
- Asynchronous transfer mode (ATM) networks accessed through:
  - LAN emulation
  - Native ATM
- Enterprise Extender (EE) networks

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for examples of the definition statements used in the XCA major node.

LAN and ATM LAN Emulation Connections

VTAM attaches to LANs through an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter, or through an IP network for Enterprise Extender. VTAM attaches to ATM networks through LAN emulation access through an IBM S/390 Open Systems Adapter. VTAM can communicate with the following types of nodes over LAN and ATM LAN emulation connections:

- Subarea

  Subarea nodes are nodes with a PU type of 4 or 5. Connections to subarea nodes are nonswitched, DIAL=NO.
- Peripheral

  Peripheral nodes are nodes with a PU type of 2 or 2.1. Connections to peripheral nodes are switched, DIAL=YES.

Connections to Subarea Nodes, DIAL=NO

To define a subarea external communication adapter major node, code the VBUILD and PORT definition statements, followed by the GROUP definition statement, and LINE and PU definition statements as pairs in the nonswitched line group.

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 143 for details on the VBUILD definition statement.

PORT Definition Statement: See “PORT Definition Statement” on page 143 for details on the PORT definition statement.

GROUP Definition Statement: Code a GROUP definition statement to group the similar characteristics of peer subarea nodes (specified with DIAL=NO).

Code a separate GROUP definition statement to describe each set of characteristics.

The major node can contain different link groups, and each link group must have a unique GROUP definition statement and contain a unique set of minor node definition statements.

LINE Definition Statement:

Code a LINE definition statement to define a single line to a peer processor or station on a LAN attached through an IBM 3172 Nways interconnect controller or through an IP network for Enterprise Extender. If the device has more than one line connected, code a separate LINE definition statement for each line.

PU Definition Statement:

Code a PU definition statement to define the physical unit associated with a line if the line defines a SNA resource. USER=SNA on the LINE definition statement defines SNA resources. Code only one PU definition statement for each LINE definition statement.

Note: If USER=VCNS is specified on the LINE definition statement, you cannot code a PU definition statement.

Table 17. External Communication Adapter Major Node - LAN, ATM LAN Emulation, or Enterprise Extender Connections, Subarea Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=XCA	184	R			
		CONFGDS	159				
		CONFGPW	159				CONFGDS
name	PORT	ADAPNO	154	R			MEDIUM
		CUADDR	161	R			
		MEDIUM	172	R			
		CAPACITY	158			8K or 4M	APPN only; VNNAME; VNGROUP
		COSTBYTE	160			0	APPN only; VNNAME; VNGROUP

Table 17. External Communication Adapter Major Node - LAN, ATM LAN Emulation, or Enterprise Extender Connections, Subarea Node (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		COSTTIME	160			0	APPN only; VNNAME; VNGROUP
		DELAY	161			0	
		PDELAY	176			TERRESTR	APPN only; VNNAME; VNGROUP
		SAPADDR	179			4	USER=SNA
		SECURITY	180			UNSECURE	APPN only; VNNAME; VNGROUP
		TGP	183				APPN only; VNNAME; VNGROUP
		TIMER	184			30	
		UPARM1	185			128	APPN only; VNNAME; VNGROUP
		UPARM2	185			128	APPN only; VNNAME; VNGROUP
		UPARM3	185			128	APPN only; VNNAME; VNGROUP
		VNGROUP	187				VNNAME; DIAL=YES
		VNNAME	187				VNGROUP; DIAL=YES
name	GROUP	DIAL	162			NO	
		SPAN	180				NetView
name	LINE	ISTATUS	170		G	ACTIVE	
		SPAN	180				NetView
		USER	186			SNA	DIAL=NO
name	PU	ISTATUS	170		G	ACTIVE	
		MACADDR	172	R			
		MULTGSUP	174			YES	
		PUTYPE	178		G, L	5	USER=VCNS
		SAPADDR	179			4	USER=SNA
		SPAN	180				NetView
		SUBAREA	181			*in_xid_value*	
		TGN	182			1	
		XNETALS	190			XNETALS start option value	

Table 17. External Communication Adapter Major Node - LAN, ATM LAN Emulation, or Enterprise Extender Connections, Subarea Node (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Connections to Peripheral Nodes, DIAL=YES

To define a peripheral external communication adapter node, code the VBUILD and PORT definition statements followed by the GROUP definition statement, and LINE and PU definition statements as pairs in the switched line group.

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 143 for details on the VBUILD definition statement.

PORT Definition Statement: See “PORT Definition Statement” on page 143 for details on the PORT definition statement.

GROUP Definition Statement: Code a GROUP definition statement to define a line group for type 2 or 2.1 nodes.

The major node can contain different link groups, and each link group must have a unique GROUP definition statement and contain a unique set of minor node definition statements.

LINE Definition Statement: Code a LINE definition statement for each switched line. The LINE definition statement defines to VTAM the name of the line and certain procedural options to be used for this line.

PU Definition Statement: Code a PU definition statement for each peripheral node with which VTAM will communicate over this switched link.

Enterprise Extender Connections

Connections to Subarea and Peripheral Nodes

VBUILD Definition Statement: Code a VBUILD definition statement for each XCA major node.

PORT Definition Statement: Code a PORT definition statement for each VBUILD TYPE=XCA definition statement. The PORT definition statement defines the VTAM connection to the Enterprise Extender shared access transport facility (SATF). Note that MEDIUM=HPRIP is required. However, only one XCA Major Node for Enterprise Extender (MEDIUM=HPRIP) can be active at a time.

VNNAME specifies the name of the virtual node representing the SATF and VNGROUP specifies the name of the DIAL=YES group for connections through the connection network.

SAPADDR=address is optional and specifies the local SAP value; the default is 4.

GROUP Definition Statement: Code a GROUP definition statement to group the similar characteristics of peer nodes in the IP network. Code a separate GROUP definition statement to describe each set of characteristics.

Note: DIAL=YES is required for Enterprise Extender.

LINE Definition Statement: Code a LINE definition statement to define a single line to a peer node in the IP network. The number of LINE statements should equal the maximum number of simultaneous connections.

Table 18. External Communication Adapter Major Node—LAN, ATM LAN Emulation, or Enterprise Extender Connections, Peripheral Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=XCA	184	R			
		CONFGDS	159				
		CONFGPW	159				CONFGDS
		VTAMTOPO	188			REPORT	
name	PORT	ADAPNO[1]	154	R			MEDIUM
		CUADDR[1]	161	R			
		MEDIUM	172	R			
		CAPACITY	158			8K	APPN only; VNNAME; VNGROUP
		COSTBYTE	160			0	APPN only; VNNAME VNGROUP
		COSTTIME	160			0	APPN only; VNNAME; VNGROUP
		IPPORT	169			12000	MEDIUM=HPRIP
		IPTOS	169			20,40,80,C0,C0	MEDIUM=HPRIP
		LIVTIME	171			10	MEDIUM=HPRIP
		PDELAY	176			TERRESTR	APPN only; VNNAME; VNGROUP
		SAPADDR	179			4	
		SRQRETRY	181			3	MEDIUM=HPRIP
		SRQTIME	181			15	MEDIUM=HPRIP
		SECURITY	180			UNSECURE	APPN only; VNNAME; VNGROUP
		TGP	183				APPN only; VNNAME; VNGROUP
		TIMER	184			30	
		UPARM1	185			128	APPN only; VNNAME; VNGROUP
		UPARM2	185			128	APPN only; VNNAME; VNGROUP

Table 18. External Communication Adapter Major Node—LAN, ATM LAN Emulation, or Enterprise Extender Connections, Peripheral Node (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		UPARM3	185			128	APPN only; VNNAME; VNGROUP
		VNGROUP	187				VNNAME; DIAL=YES
		VNNAME	187				VNGROUP; DIAL=YES
name	GROUP	DIAL	162	R		NO	DIAL=YES
		AUTOGEN	156				
		DYNPU	167			NO	
		DYNPUPFX	168			CN	DYNPU=YES
		SPAN	180				NetView
		VTAMTOPO	188				
name	LINE	ANSWER	155		G	ON	CALL=OUT
		CALL	157		G	IN	
		ISTATUS	170		G	ACTIVE	
		LIMRES	170		G	NO	LIMQSINT; LU 6.2; PUTYPE=1, 2
		SPAN	180				NetView
		SWNORDER	182		G	CPNAME	
		VTAMTOPO	188		G		
name	PU	ISTATUS	170		G	ACTIVE	
		SPAN	180				NetView
		XNETALS	190			XNETALS start option value	

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Note: [1]For LAN or ATM emulation connections only.

ATM Native Connections

VTAM attaches to ATM networks through native ATM access through an IBM S/390 Open Systems Adapter. VTAM can communicate with HPR APPN nodes over ATM native connections. Communication occurs over the following types of connections:

- Transmission groups (TGs) over permanent virtual channels (PVCs), DIAL=NO
- TGs over switched virtual channels (SVCs), DIAL=YES
- TGs to connection networks, DIAL=YES

To define an external communication adapter major node for an ATM native connection, code the VBUILD and PORT definition statements followed by GROUP, LINE, and PU definition statements to define the TGs.

VBUILD Definition Statement

See “VBUILD Definition Statement” on page 143 for details on the VBUILD definition statement.

PORT Definition Statement

See “PORT Definition Statement” on page 143 for details on the PORT definition statement.

Table 19. External Communication Adapter Major Node—ATM Native Connection, Port

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=XCA	184	R			
		CONFGDS	159				
		CONFGPW	159				CONFGDS
		VTAMTOPO	188				DIAL=YES
name	PORT	PORTNAME	177	R			
		MEDIUM	172			ATM	PORTNAME
		SAPADDR	179			4	

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Transmission Groups (TGs) Over Permanent Virtual Channels (PVCs), DIAL=NO

To define TGs over PVCs, code a GROUP definition statement, followed by one LINE and one PU definition statement, for each PVC to which you subscribe through the port defined by this XCA major node.

GROUP Definition Statement: Code a GROUP definition statement to define a line group for remote nodes connected through the port defined by this XCA major node.

LINE Definition Statement: Code a LINE definition statement for each PVC to which you subscribe through the port defined by this XCA major node. The LINE definition statement defines to VTAM the name of the PVC and certain procedural options to be used for the PVC.

PU Definition Statement: Code a PU definition statement for a remote node with which VTAM can communicate over a PVC. The PU definition statement should follow the LINE definition statement that defines the PVC through which the remote node is connected.

External Communication Adapter Major Node

Table 20. External Communication Adapter Major Node—ATM Native Connection, TGs over PVCs

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	GROUP	DIAL=NO	162	R			nonswitched lines
		SPAN	180				NetView
name	LINE	PVCNAME	178	R			
		ISTATUS	170		G	ACTIVE	
		SPAN	180				NetView
		USER	186			SNA	
name	PU	AUTHLEN	155			YES	PU type 2.1
		CAPACITY	158			8K	
		CONNTYPE	159			APPN	
		COSTBYTE	160			0	
		COSTTIME	160			0	
		CPCP	160			CPCP start option	
		CPNAME	161				
		DYNADJCP	166			DYNADJCP start option	CPNAME
		DYNLU	166			DYNLU start option	
		HPR	169			YES	
		ISTATUS	170		G	ACTIVE	
		NATIVE	174				BN=YES
		NETID	175				XNETALS
		NN	176				
		PDELAY	176			TERRESTR	
		PUTYPE	178			2	
		SAPADDR	179			4	
		SECURITY	180			UNSECURE	
		SPAN	180				NetView
		TGN	182			ANY	
		TGP	183				
		TOPO	184			CONNECT	
		UPARM1	185			128	
		UPARM2	185			128	
		UPARM3	185			128	
		VERALSID	186			NO	
		XID	190			YES	
		XNETALS	190			XNETALS start option value	

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Transmission Groups (TGs) Over Switched Virtual Channels (SVCs), DIAL=YES

To define TGs over SVCs, code a GROUP definition statement, followed by LINE and PU definition statements for the connections you expect to have over the SVCs through the port defined by this XCA major node. The LINE definition statements serve as placeholders for lines used to connect to remote nodes that are defined either dynamically, or in a switched major node, and are used when those nodes are activated.

GROUP Definition Statement: Code a GROUP definition statement to define a line group for remote nodes connected through the port defined by this XCA major node.

LINE Definition Statement: Code LINE definition statements for the connections you expect to have over the SVCs through the port defined by this XCA major node. The total number of LINE definition statements should equal the maximum number of simultaneous connections you can have. The LINE definition statement defines to VTAM certain procedural options to be used for an SVC assigned to this line.

PU Definition Statement: Code a PU definition statement for a remote node with which VTAM can communicate over the SVCs through the port defined by this XCA major node.

Table 21. External Communication Adapter Major Node—ATM Native Connection, TGs over SVCs

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	GROUP	DIAL=YES	162	R		NO	switched lines
		AUTOGEN	156				
		DYNPU	167			NO	
		DYNPUPFX	168			CN	DYNPU=YES
		SPAN	180				NetView
		VTAMTOPO	188				DIAL=YES
name	LINE	ANSWER	155		G	ON	
		CALL	157		G	IN	
		ISTATUS	170		G	ACTIVE	
		LIMRES	170		G	NO	LU 6.2
		SPAN	180				NetView
		SWNORDER	182		G	CPNAME	
		VTAMTOPO	188				DIAL=YES
name	PU	ISTATUS	170		G	ACTIVE	
		SPAN	180				NetView

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Transmission Groups (TGs) to Connection Networks, DIAL=YES

To define a TG to a connection network, code a GROUP definition statement, followed by LINE and PU definition statements for the connections you expect to have using the connection network TG through the port defined by this XCA major node. The LINE definition statements serve as placeholders for SVCs to remote nodes that are defined either dynamically, or in a switched major node, and are used when those nodes are activated.

GROUP Definition Statement: Code a GROUP definition statement to define a TG to a connection network through the port defined by this XCA major node.

LINE Definition Statement: Code LINE definition statements for the connections you expect to have over the connection network TG through the port defined by this XCA major node. The total number of LINE definition statements should equal the maximum number of simultaneous connections you can have using the connection network TG. The LINE definition statement defines to VTAM certain procedural options to be used for a connection network TG assigned to this line.

PU Definition Statement: Code a PU definition statement for a remote node with which VTAM can communicate over the connection network TG through the port defined by this XCA major node.

Table 22. External Communication Adapter Major Node - ATM Native Connection, TGs to Connection Networks

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	GROUP	DIAL=YES	162	R		NO	switched lines
		AUTOGEN	156				
		CAPACITY	158			25M	
		COSTBYTE	160			0	
		COSTTIME	160			0	
		DLCADDR	162				
		DYNPU	167			NO	
		DYNPUPFX	168			CN	DYNPU=YES
		PDELAY	176			NEGLIGLB	
		SECURITY	180			UNSECURE	
		SPAN	180				NetView
		TGP	183				
		UPARM1	185			128	
		UPARM2	185			128	
		UPARM3	185			128	
		VTAMTOPO	188				DIAL=YES
name	LINE	ANSWER	155		G	ON	
		CALL	157		G	IN	
		ISTATUS	170		G	ACTIVE	
		LIMRES	170		G	NO	LU 6.2
		SPAN	180				NetView
		SWNORDER	182		G	CPNAME	
		VTAMTOPO	188				DIAL=YES
name	PU	ISTATUS	170		G	ACTIVE	
		SPAN	180				NetView

Table 22. External Communication Adapter Major Node - ATM Native Connection, TGs to Connection Networks (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each XCA major node.

PORT Definition Statement

Code one PORT definition statement for each VBUILD TYPE=XCA definition statement. The PORT definition statement defines the VTAM connection to a shared access transport facility, such as:

- A CSMA/CD 802.3 LAN, a token-ring network, or a FDDI LAN attached through an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter, or through an IP network for Enterprise Extender.
- An ATM network attached through an IBM S/390 Open Systems Adapter through LAN emulation access

  Note: ATM networks attached through an IBM S/390 Open Systems Adapter through LAN emulation access are defined to VTAM as though they are CSMA/CD 802.3 LANs or token-ring networks.
- An ATM network attached through an IBM S/390 Open Systems Adapter through native ATM access.

The PORT definition statement also defines the actual adapter on the IBM 3172 Nways Interconnect Controller, IBM S/390 Open Systems Adapter, or through an IP network for Enterprise Extender through which the connection is made.

Full Syntax

The full syntax for the external communication adapter (XCA) major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

LAN and ATM LAN Emulation Connections

VBUILD:

```
►►──name──VBUILD──TYPE=XCA──┬──────────────────────────────────────────────┬──►
                            ├─CONFGDS=confgds_name─────────────────────────┤
                            └─CONFGPW=confgds_password,CONFGDS=confgds_name─┘
```

External Communication Adapter Major Node

```
►─┬─VTAMTOPO=REPORT──────────┬──────────────────────────────────────►◄
  │                          │
  └─VTAMTOPO=─┬─REPORT───────┘
              │
              ├─NOREPORT─────
              ├─NOLLINES─────
              ├─NOSWPUS──────
              ├─INCLUDE──────
              └─IGNORE───────
```

PORT:

```
►►─name──PORT──ADAPNO=adapter_number──CUADDR=device_address──────────────►

                                   (1)
                        ┌─CAPACITY=4M──────────────┐
                        ├─CAPACITY=8K──────────────┤   ┌─COSTBYTE=0──────────────┐
►─MEDIUM=──┬─ATM──────┼──────────────────────────┼───┼─────────────────────────┼──►
           ├─BOXMGR───┤  └─CAPACITY=number_of_bits─┘   └─COSTBYTE=cost_per_byte──┘
           ├─CSMACD───┤
           ├─FDDI─────┤
           └─LONG─────┘
  ┌─COSTTIME=0──────────────────┐   ┌─DELAY=0──────────────┐
►─┼─────────────────────────────┼───┼──────────────────────┼──────────────►
  └─COSTTIME=cost_per_unit_time─┘   └─DELAY=timeout_value──┘

                   (2)
  ┌─PDELAY=NEGLIGIB──────┐
  ├─PDELAY=TERRESTR──────┤   ┌─SAPADDR=4──────────┐   ┌─SECURITY=UNSECURE──────┐
►─┼──────────────────────┼───┼────────────────────┼───┼────────────────────────┼──►
  │                      │   └─SAPADDR=address────┘   │                        │
  └─PDELAY=─┬─TERRESTR───┤                            └─SECURITY=─┬─UNSECURE────┤
            │            │                                        │             │
            ├─LONG───────┤                                        ├─PUBLIC──────┤
            ├─NEGLIGIB───┤                                        ├─UNDERGRO────┤
            └─PACKET─────┘                                        ├─SECURE──────┤
                                                                  ├─GUARDED─────┤
                                                                  ├─ENCRYPT─────┤
                                                                  └─SHIELDED────┘
                          ┌─TIMER=30──────────────┐
►─┬──────────────────────┼───────────────────────┼──────────────────────────►
  └─TGP=tgp_name─────────┘   └─TIMER=timeout_value─┘

  ┌─UPARM1=128──────────────────────┐   ┌─UPARM2=128──────────────────────┐
►─┼─────────────────────────────────┼───┼─────────────────────────────────┼──►
  └─UPARM1=user-defined_operand_value┘   └─UPARM2=user-defined_operand_value┘

  ┌─UPARM3=128──────────────────────┐
►─┼─────────────────────────────────┼──────────────────────────────────────►
  └─UPARM3=user-defined_operand_value┘
```

```
►──┬──────────────
```

External Communication Adapter Major Node

```
►┬─SWNORDER=value_of_swnorder_start_option──────┬──┬─USER=SNA──────┬──────────►
 │                                              │  │               │
 └─SWNORDER=┬─value_of_swnorder_start_option─┬┘  └─USER=┬─SNA─┬┬┘
            ├─CPNAME──────────────────────────┤          └─VCNS┘│
            └─STATNID─────────────────────────┘               │

►┬──────────────────────────────────────────────────────────────────────────►◄
 │                 ┌─REPORT───┐
 └─VTAMTOPO=┬──────┤          ├┘
            └─NOREPORT─────────┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

PU:

```
►►─name──PU──MACADDR=macaddr──┬─ISTATUS=ACTIVE──────┬──┬─PUTYPE=5──────┬──────►
                              │                     │  │               │
                              └─ISTATUS=┬─ACTIVE───┬┘  └─PUTYPE=┬─5─┬┬┘
                                        └─INACTIVE─┘            └─4─┘│
                                                                     │

►┬─MULTGSUP=YES──────┬──┬─SAPADDR=4──────────┬──┬──────────────────┬──────────►
 │                   │  │                    │  │                  │
 └─MULTGSUP=┬─YES─┬┬┘  └─SAPADDR=address────┘  └─SPAN=span_name────┘
            └─NO──┘│
                   │

►┬─SUBAREA=inbound_xid_value─┬──┬─TGN=1──────────────┬──────────────────────────►
 │                           │  │                    │
 └─SUBAREA=suba_addr─────────┘  └─TGN=┬─1──────────┬┬┘
                                      ├─ANY────────┤│
                                      └─tg_number──┘│
                                                    │

►┬─XNETALS=value_of_XNETALS_start_option─┬──────────────────────────────────►◄
 │                                       │
 └─XNETALS=┬─NO──┬──────────────────────┬┘
           └─YES─┘                      │
```

ATM Native Connections

VBUILD:

```
►►─name──VBUILD──TYPE=XCA──┬─────────────────────────────────────────────────►
                           ├─CONFGDS=confgds_name──────────────────────────────┤
                           └─CONFGPW=confgds_password,CONFGDS=confgds_name──────┘
```

```
►─┬─VTAMTOPO=REPORT──────────┬──────────────────────────────────────►◄
  │                          │
  └─VTAMTOPO=─┬─REPORT───────┬─┘
              │              │
              ├─NOREPORT─────┤
              ├─NOLLINES─────┤
              ├─NOSWPUS──────┤
              ├─INCLUDE──────┤
              └─IGNORE───────┘
```

PORT:

```
►►─name──PORT──┬─MEDIUM=ATM─┬──PORTNAME=(port_name,link_num)──────────►
               │            │
               └─MEDIUM=ATM─┘

►─┬─SAPADDR=4────────────┬──────────────────────────────────────────►◄
  │                      │
  └─SAPADDR=address──────┘
```

GROUP:

```
►►─name──GROUP──────────────────────────────────────────────────────►

►─┬──────────────────────────────────────────────────────────────────►
  │
  └─AUTOGEN=─┬─num_stmts──────────────────────────────────────────────┬
             │                                                        │
             └─(num_stmts─┬─,0──────────────────┬─┬─,Q──────────────┬─┬─)─┘
                          │                     │ │                 │ │
                          └─,line_seed_char─────┘ └─,pu_seed_char───┘ │

►─┬─COSTBYTE=0──────────────┬─┬─COSTTIME=0──────────────────┬──────────►
  │                         │ │                             │
  └─COSTBYTE=cost_per_byte──┘ └─COSTTIME=cost_per_unit_time─┘

►─┬─DIAL=NO──────────┬──┬──────────────────────────────────────────►
  │                  │  │
  └─DIAL=─┬─NO───────┬─┘  └─DLCADDR=(─subfield_id,type,─string──)──┘
          │          │
          └─YES──────┘

►─┬─DYNPU=NO─────────┬─┬─DYNPUPFX=CN──────────┬─┬─PDELAY=NEGLIGIB──────────┬──────────►
  │                  │ │                      │ │                          │
  └─DYNPU=─┬─NO───────┬─┘ └─DYNPUPFX=string────┘ └─PDELAY=─┬─NEGLIGIB─────────┬─┘
           │          │                                   │                  │
           └─YES──────┘                                   ├─LONG─────────────┤
                                                          ├─PACKET───────────┤
                                                          └─TERRESTR─────────┘
```

External Communication Adapter Major Node

```
►─┬─SECURITY=UNSECURE──────────┬──┬──────────────┬──┬──────────────┬──►
  │                            │  └─SPAN=span_name─┘  └─TGP=tgp_name─┘
  └─SECURITY=─┬─UNSECURE──┬────┘
              ├─PUBLIC────┤
              ├─UNDERGRO──┤
              ├─SECURE────┤
              ├─GUARDED───┤
              ├─ENCRYPT───┤
              └─SHIELDED──┘

►─┬─UPARM1=128──────────────────────┬─┬─UPARM2=128──────────────────────┬──►
  └─UPARM1=user-defined_operand_value─┘ └─UPARM2=user-defined_operand_value─┘

►─┬─UPARM3=128──────────────────────┬──┬──────────────────────────────┬──►◄
  └─UPARM3=user-defined_operand_value─┘  └─VTAMTOPO=─┬─REPORT───┬────┘
                                                     ├─NOREPORT─┤
                                                     ├─NOLLINES─┤
                                                     └─NOSWPUS──┘
```

LINE:

```
►►─name──LINE─┬─ANSWER=ON──────┬─┬─CALL=IN──────────┬──►
              │                │ │                  │
              └─ANSWER=─┬─ON──┬─┘ └─CALL=─┬─IN────┬─┘
                        └─OFF─┘           ├─INOUT──┤
                                          └─OUT────┘

►─┬─ISTATUS=ACTIVE──────────┬─┬─LIMRES=NO──────────┬─┬──────────────┬──►
  │                         │ │                    │ └─PVCNAME=pvc_name─┘
  └─ISTATUS=─┬─ACTIVE───┬───┘ └─LIMRES=─┬─NO──┬────┘
             └─INACTIVE─┘               └─YES─┘
             │ ┌─LIMRES=NO──┬─(1)
             │ └─LIMRES=NO──┘
             │
             │
             │

►─┬──────────────┬─┬─SWNORDER=value_of_swnorder_start_option──────────┬──►
  └─SPAN=span_name─┘ │                                                 │
                     └─SWNORDER=─┬─value_of_swnorder_start_option──┬──┘
                                 ├─CPNAME──────────────────────────┤
                                 └─STATNID─────────────────────────┘
```

```
►─┬─USER=SNA──────────┬──┬──────────────────────────────────────────────────────────────►◄
  │                   │  │
  └─USER=─┬─SNA─┬─────┘  └─VTAMTOPO=─┬─REPORT───┬──────────────────────────────────────────
          └─VCNS┘                   └─NOREPORT─┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

PU:

```
►►─name─PU─┬─AUTHLEN=YES──────────┬─┬─CONNTYPE=APPN─┬──────────────────────────────────►
           │                      │ │               │
           └─AUTHLEN=─┬─NO─┬──────┘ └─CONNTYPE=APPN─┘
                      └─YES┘

►─┬─COSTBYTE=0──────────────────┬─┬─COSTTIME=0──────────────────────────┬──────────────►
  │                             │ │                                     │
  └─COSTBYTE=cost_per_byte──────┘ └─COSTTIME=cost_per_unit_time─────────┘

►─┬─CPCP=value_of_cpcp_start_option──────────┬─┬──────────────────────────────────────►
  │                                          │ │
  │                  ┌─value_of_cpcp_start_option─┐ └─CPNAME=control_point_name──┘
  └─CPCP=────────────┼─NO────────────────────────┤
                     └─YES───────────────────────┘

►─┬─DYNLU=value_of_dynlu_start_option──────────┬──────────────────────────────────────►
  │                                            │
  │                  ┌─value_of_dynlu_start_option─┐
  │  └─DYNLU=────────┼─NO──────────────────────────┤
  │                  └─YES─────────────────────────┘
  │  ┌─DYNLU=YES─┐  (1)
  │  │           │
  │  └─DYNLU=YES─┘
  │
  │
  │
```

External Communication Adapter Major Node

```
►─┬─ISTATUS=ACTIVE──────────┬──┬──────────────────────┬──┬──────────────────────┬──►
  │                         │  │                      │  │                      │
  └─ISTATUS=─┬─ACTIVE───┬───┘  └─NATIVE=─┬─NO───┬───┘  └─NETID=network_id───┘
             │           │               └─YES──┘
             └─INACTIVE──┘

►─┬──────────────────────┬──┬─┬─PDELAY=NEGLIGIB──────────┬─┬─PUTYPE=2─┬──►
  │                      │  │ │                          │ │          │
  └─NN=─┬─NO───┬──────┬──┘  │ └─PDELAY=─┬─NEGLIGIB─┬───┘ └──────────┘
        └─YES──┘      │     │           │          │
                      │     │           ├─LONG─────┤
                      │     │           ├─PACKET───┤
                      │     │           └─TERRESTR─┘

►─┬─SAPADDR=4──────────┬──────────────────────────────────────────────────►
  │                    │
  └─SAPADDR=address────┘

►─┬─SECURITY=UNSECURE──────────┬──┬──────────────────────┬──┬─TGN=ANY──────────┬──►
  │                            │  │                      │  │                  │
  └─SECURITY=─┬─UNSECURE─┬───┬──┘  └─SPAN=span_name───┘  └─TGN=─┬─ANY──────┬──┘
              │          │   │                                  │          │
              ├─PUBLIC───┤   │                                  └─tg_number┘
              ├─UNDERGRO─┤   │
              ├─SECURE───┤   │
              ├─GUARDED──┤   │
              ├─ENCRYPT──┤   │
              └─SHIELDED─┘   │

►─┬──────────────────────┬──┬──────────────────────┬──┬─UPARM1=128──────────────────────┬──►
  │                      │  │                      │  │                                 │
  └─TGP=tgp_name─────────┘  └─TOPO=CONNECT─────────┘  └─UPARM1=user-defined_operand_value─┘

►─┬─UPARM2=128──────────────────────┬─┬─UPARM3=128──────────────────────┬──►
  │                                 │ │                                 │
  └─UPARM2=user-defined_operand_value┘ └─UPARM3=user-defined_operand_value┘

►─┬─VERALSID=NO──────────┬─┬─XID=NO──────────┬──────────────────────────────►
  │                      │ │                 │
  └─VERALSID=─┬─NO───┬───┘ └─XID=─┬─NO───┬───┘
              └─YES──┘            └─YES──┘
  │ ┌─VERALSID=NO─┬─(3)
  │ │             │
  └─┴─VERALSID=NO─┘

►─┬─XNETALS=value_of_XNETALS_start_option─┬──────────────────────────────►◄
  │                                       │
  └─XNETALS=─┬─NO───┬──────────────────────┘
             └─YES──┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

2 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

3 VERALSID=NO is the only valid value if the TRLE operand is coded.

Enterprise Extender Predefined Connections

VBUILD:

```
►►──name──VBUILD──TYPE=XCA──┬──────────────────────────────────────────────────►
                            ├─CONFGDS=confgds_name──────────────────────────────┤
                            └─CONFGPW=confgds_password,CONFGDS=confgds_name─────┘

  ┌─VTAMTOPO=REPORT──────┐
►─┤                      ├──────────────────────────────────────────────────►◄
  └─VTAMTOPO=─┬─REPORT───┬─┘
              ├─NOREPORT─┤
              ├─NOLLINES─┤
              ├─NOSWPUS──┤
              ├─INCLUDE──┤
              └─IGNORE───┘
```

PORT:

```
►►──name──PORT──────────────────────────────────────────────────────────────►

  ┌─IPTOS=(20, 40, 80, C0,C0)──────────────────────────────────────────────┐
►─┤                                                                        ├──►
  │                                                                        │
  └─IPTOS=(─┬─20──────┬─,─┬─40──────┬─,─┬─80──────┬─,─┬─C0──────────┬─,─┬─C0──────────┬─)─┘
            └─low─────┘   └─med─────┘   └─high────┘   └─network─────┘   └─signal────┘

  ┌─LIVTIME=10──────┐                  ┌─IPPORT=12000──────┐  ┌─SAPADDR=4──────────┐
►─┤                 ├─MEDIUM=HPRIP─────┤                   ├──┤                    ├──────►
  └─LIVTIME=time────┘                  └─IPPORT=number─────┘  └─SAPADDR=address────┘

  ┌─SRQRETRY=3──────────────────┐  ┌─SRQTIME=15──────┐
►─┤                             ├──┤                 ├──────────────────────────►◄
  └─SRQRETRY=number_of_retries──┘  └─SRQTIME=time────┘
```

GROUP:

```
►►──name──GROUP─────────────────────────────────────────────────────────────►
```

External Communication Adapter Major Node

```
►──┬──────────────────────────────────────────────────────────────────────────►
   └AUTOGEN=──┬─num_stmts──────────────────────────────────────────────────────┘
              │
              └(num_stmts──┬─,0──────────────────┬──┬─,Q──────────────────┬──)─┘
                           │                     │  │                     │
                           └─,line_seed_char─────┘  └─,pu_seed_char───────┘

►─DIAL=YES──┬─DYNPU=NO──────────┬──┬─DYNPUPFX=CN──────────┬──┬──────────────────►
            │                   │  │                      │  │
            └─DYNPU=──┬─NO──┬───┘  └─DYNPUPFX=string──────┘  └─SPAN=span_name──┘
                      │     │
                      └─YES─┘

►──┬──────────────────────────────────────────────────────────────────────────►◄
   └TGP=tgp_name──┘
```

LINE:

```
►►─name──LINE──┬─ANSWER=ON──────────┬──┬─CALL=IN──────────────┬──────────────►
               │                     │  │                      │
               └─ANSWER=──┬─ON──┬────┘  └─CALL=──┬─IN────┬────┘
                          │     │                │       │
                          └─OFF─┘                ├─INOUT─┤
                                                 └─OUT───┘

►──┬─ISTATUS=ACTIVE──────────┬──┬──────────────────────────────────────────►
   │                         │  │
   └─ISTATUS=──┬─ACTIVE───┬───┘  └─SPAN=span_name──┘
               │           │
               └─INACTIVE──┘

►──┬─SWNORDER=value_of_swnorder_start_option──────────┬──────────────────────►◄
   │                                                  │
   └─SWNORDER=──┬─value_of_swnorder_start_option──┬──┘
                │                                 │
                ├─CPNAME──────────────────────────┤
                └─STATNID─────────────────────────┘
```

Enterprise Extender Connection Network Facility Connections

VBUILD:

```
►►─name──VBUILD──TYPE=XCA──┬──────────────────────────────────────────────────►
                            │
                            ├─CONFGDS=confgds_name──────────────────────────┤
                            └─CONFGPW=confgds_password,CONFGDS=confgds_name──┘
```

```
►─┬─VTAMTOPO=REPORT──────────┬──────────────────────────────────────►◄
  │                          │
  └─VTAMTOPO=─┬─REPORT───┬───┘
              ├─NOREPORT─┤
              ├─NOLLINES─┤
              ├─NOSWPUS──┤
              ├─INCLUDE──┤
              └─IGNORE───┘
```

PORT:

```
►►─name──PORT──────────────────────────────────────────────────────►

►─┬─IPTOS=(20, 40, 80, C0,C0)──────────────────────────────────────┬─►
  │                                                                │
  └─IPTOS=(─┬─20──────┬─,─┬─40──────┬─,─┬─80──────┬─,─┬─C0──────────┬─,─┬─C0──────────┬─)─┘
            └─low─────┘   └─med─────┘   └─high────┘   └─network─────┘   └─signal────┘

►─┬─LIVTIME=10──────┬──MEDIUM=HPRIP──┬─IPPORT=12000──────┬──┬─SAPADDR=4──────────┬──────►
  └─LIVTIME=time────┘                └─IPPORT=number─────┘  └─SAPADDR=address────┘

►─┬─SRQRETRY=3──────────────────┬──┬─SRQTIME=15──────┬──────────────────────────────────►
  └─SRQRETRY=number_of_retries──┘  └─SRQTIME=time────┘

►─┬─────────────────────────────────────────────────────────────────────────────────────►◄
  └─VNNAME=connection_network_cpname,VNGROUP=group_name──────────────────────────────────┘
```

GROUP:

```
►►─name──GROUP─────────────────────────────────────────────────────►

►─┬─────────────────────────────────────────────────────────────────────────────────────►
  │                                                                                       │
  └─AUTOGEN=─┬─num_stmts──────────────────────────────────────────────────────────────────┬─┘
             │                                                                            │
             └─(num_stmts─┬─,0──────────────────┬──┬─,Q──────────────────┬──)──────────────┘
                          └─,line_seed_char─────┘  └─,pu_seed_char───────┘

►─DIAL=YES─┬─DYNPU=NO────────┬──┬─DYNPUPFX=CN──────┬──┬──────────────────────────────────►
           │                 │  └─DYNPUPFX=string───┘  └─SPAN=span_name──────────────────┘
           └─DYNPU=─┬─NO──┬────┘
                    └─YES─┘
```

External Communication Adapter Major Node

```
►──┬──────────────────┬──────────────────────────────────────────────────►◄
   └─TGP=tgp_name─────┘
```

LINE:

```
                   ┌─ANSWER=ON──────┐   ┌─CALL=IN────────┐
►►──name──LINE─────┼────────────────┼───┼────────────────┼──────────────►
                   │            ┌─ON─┐ │       ┌─IN────┐ │
                   └─ANSWER=────┼────┼─┘       └─CALL=──┼───────┼─┘
                                └─OFF┘                  ├─INOUT─┤
                                                        └─OUT───┘

   ┌─ISTATUS=ACTIVE─────────┐
►──┼────────────────────────┼──┬──────────────────────┬──────────────►
   │              ┌─ACTIVE─┐ │  └─SPAN=span_name───────┘
   └─ISTATUS=─────┼────────┼─┘
                  └─INACTIVE┘

   ┌─SWNORDER=value_of_swnorder_start_option──────┐
►──┼──────────────────────────────────────────────┼──────────────────────►◄
   │                 ┌─value_of_swnorder_start_option─┐
   └─SWNORDER=───────┼────────────────────────────────┼─┘
                     ├─CPNAME─────────────────────────┤
                     └─STATNID────────────────────────┘
```

Operand Descriptions

The following section describes the operands you can code for the external communication adapter (XCA) major node.

ADAPNO

```
►►──ADAPNO=adapter_number──────────────────────────────────────────────►◄
```

statements:	*PORT*
dependencies:	*LAN and ATM LAN emulation connections only, MEDIUM*
range:	*0–127*

Specifies a relative adapter number assigned to each adapter or port by an IBM 3172 Nways Interconnect Controller or an IBM S/390 Open Systems Adapter.

adapter_number is a 1–3 digit decimal integer in the range 0–127. This operand is used in conjunction with the MEDIUM operand to uniquely define an adapter or port within an IBM 3172 Nways Interconnect Controller or an IBM S/390 Open Systems Adapter.

This operand is required for LAN and ATM LAN emulation connections.

ANSWER

```
►►─┬─ANSWER=ON──────────┬──────────────────────────────────────────►◄
   │                    │
   └─ANSWER=─┬─ON──┬────┘
             └─OFF─┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	GROUP, LINE	*statements:*	GROUP, LINE
dependencies:	Peripheral node CALL=IN or INOUT	*dependencies:*	TGs over SVCs TGs to connection networks DIAL=YES, CALL=IN or INOUT

Specifies whether physical units can dial in to VTAM.

ANSWER=OFF
Physical units cannot dial in to VTAM, regardless of whether the line is active or inactive.

ANSWER=ON
Physical units can dial in to VTAM when the line is activated.

The operator can issue a VARY ANS command after activation to change the answer state of the line. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ANS command.

Note: When CALL=OUT, the ANSWER operand is ignored, and the operator receives a message that this operand is not valid.

AUTHLEN

```
►►─┬─AUTHLEN=YES────────┬──────────────────────────────────────────►◄
   │                    │
   └─AUTHLEN=─┬─NO──┬───┘
              └─YES─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies whether VTAM should pass the transmission priority field specified by this PU to another PU.

AUTHLEN=NO
VTAM will not pass the specified transmission priority; the default priority of medium will be passed.

AUTHLEN=YES
VTAM will pass the specified transmission priority.

AUTOGEN

```
►►──┬──────────────────────────────────────────────────────────────────────────────┬──►◄
    └AUTOGEN=──┬─num_stmts──────────────────────────────────────────────────────────┬
               │                                                                    │
               └(num_stmts──┬─,0──────────────────┬──┬─,Q──────────────────┬──)────┘
                            │                     │  │                     │
                            └─,line_seed_char─────┘  └─,pu_seed_char───────┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	GROUP	*statements:*	GROUP
dependencies:	Peripheral node	*dependencies:*	TGs over SVCs TGs to connection networks DIAL=YES

Defines LINE and PU definition statements that VTAM automatically generates.

num_stmts
Specifies the number of VTAM-generated LINE and PU statements. Code a decimal value in the range 1–4096. This value is required.

Note: Interconnect Controller'.The range is 1–255. If OF/2 is being used, the range is 1–245.

For ATM Native Connections: should be the number of concurrent SVCs that you expect to have activated.

line_seed_char
Defines the first 1–5 characters of the name that VTAM creates for each LINE statement VTAM automatically generates.

pu_seed_char
Defines the first 1–5 characters of the name that VTAM creates for each PU statement VTAM automatically generates.

Note: If you do not want to specify *line_seed_character*, code a comma in its place. If you do not want to specify *pu_seed_character*, code a comma in its place.

VTAM uses the following naming convention to create a LINE definition statement name and a PU definition statement name:

```
sssssnnn
```

where:

sssss
Is the value specified by *line_seed_char* or *pu_seed_char*.

– If you specify a single-character seed value, VTAM adds the device number coded on the CUADDR operand on the PORT definition statement in the XCA major node to the single character seed value, then adds a sequential number to the end of the name.

For ATM Native Connections: of the device number coded on the CUADDR operand.

– If you specify a seed value greater than a single character, VTAM uses the multiple-character seed value you specify, then adds a sequential number to the end of the name.

– If you do not specify a seed character, VTAM uses the default value of Q.

nnn
Is a sequential hexadecimal number created by VTAM (X'0'–X'FFF').

Examples: The following tables show the LINE and PU definition statements and names that VTAM generates based on the AUTOGEN operand.

AUTOGEN=(3,,,) and CUA=505:

Note: Seed characters are not specified. Therefore, the default character O is used for LINE statements and the default character Q is used for PU statements.

Table 23. XCA AUTOGEN without User-Supplied Seeds

LINE	O0505000	PU	Q0505000
LINE	O0505001	PU	Q0505001
LINE	O0505002	PU	Q0505002

AUTOGEN=(2, K, J) and CUA=505:

Table 24. XCA AUTOGEN with Single-Character User-Supplied Seeds

LINE	K0505000	PU	J0505000
LINE	K0505001	PU	J0505001

AUTOGEN=(2, KLM, HIJ) and CUA=505:

Note: Device number is not included in the LINE and PU definition statement names because multiple-character seed values are specified for each.

Table 25. XCA AUTOGEN with 3-Character User-Supplied Seeds

LINE	KLM00000	PU	HIJ00000
LINE	KLM00001	PU	HIJ00001

CALL

```
►►─┬─CALL=IN──────────┬──────────────────────────────────────────►◄
   │                  │
   └─CALL=─┬─IN────┬──┘
           ├─INOUT─┤
           └─OUT───┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		**For ATM Native Connections:**	
statements:	GROUP, LINE	*statements:*	GROUP, LINE

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
dependencies:	Peripheral node	*dependencies:*	TGs over SVCs TGs to connection networks DIAL=YES

Specifies whether physical units, or VTAM, or both, can initiate calls over the line represented by this LINE definition statement.

CALL=IN
Specifies that the line is to be used only for incoming calls (the physical units are to call VTAM).

CALL=INOUT
Specifies that the line is to be used for both incoming and outgoing calls.

CALL=OUT
Specifies that the line is to be used only for outgoing calls (VTAM is to call the physical units).

CAPACITY

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
(1) ►► ┌CAPACITY=4M──────┐ ├CAPACITY=8K──────┤ └CAPACITY=*number_of_bits*──┘ ►◄ **Notes:** **1** For connections to virtual nodes, the default value is 4M.		►► ┌CAPACITY=25M──────┐ └CAPACITY=*number_of_bits*──┘ ►◄	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	1K–1000M	*range:*	1K–1000M

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either kilobits per second (for example, 100K) or megabits per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Notes:

1. Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.
2. The capacity specified for ATM native connections defined as reserved bandwidth should be related to the forward sustainable cell rate (DLCADDR subfield 7). To calculate the data throughput required for a reserved bandwidth connection, multiply the sustainable cell rate by the payload per cell (48 bytes). If the defined CAPACITY is greater than ten times the ATM reserved bandwidth connection capacity, excess storage might be utilized and the weighting of this TG with others might cause undesirable results.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └CONFGDS=name──────┘
```

statements: *VBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └CONFGPW=password──┘
```

statements: *VBUILD*
dependencies: *CONFGDS*

Specifies the 1–7 character alphanumeric password used by VTAM to gain access to the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

CONNTYPE

```
    ┌─CONNTYPE=APPN─┐
►►──┤               ├──────────────────────────────────────────────────────►◄
    └─CONNTYPE=APPN─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies that the connection is to be established as an APPN connection that can support parallel TGs, CP-CP sessions, and CP name change support. CONNTYPE=APPN is the only valid value for ATM native connections and is the default value.

COSTBYTE

```
►►─┬─COSTBYTE=0──────────────┬──────────────────────────────────────────►◄
   └─COSTBYTE=cost_per_byte──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	0–255	*range:*	0–255

Specifies a cost-per-byte-transmitted to be associated with the transmission group (TG). A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
►►─┬─COSTTIME=0──────────────┬──────────────────────────────────────────►◄
   └─COSTTIME=cost_per_unit_time──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	0–255	*range:*	0–255

Specifies a cost-per-unit-of-time to be associated with the transmission group (TG). A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

CPCP

```
   ┌─CPCP=value_of_cpcp_start_option──────┐
►►─┼──────────────────────────────────────┼──────────────────────────────────►◄
   │          ┌─value_of_cpcp_start_option─┐
   └─CPCP=────┼────────────────────────────┼──┘
              ├─NO─────────────────────────┤
              └─YES────────────────────────┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies whether CP-CP sessions are supported on this connection. If you do not code this operand, the value defaults to the value specified on the CPCP start option.

CPCP=NO
Indicates CP-CP sessions are not supported on this connection.

CPCP=YES
Indicates CP-CP sessions are supported on this connection.

CPNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─CPNAME=control_point_name─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Identifies the resource's owning control point name. If it is later learned that the owning control point name is different than what is specified here, the resource will then be associated with the newly learned control point name.

CUADDR

```
►►──CUADDR=device_address──────────────────────────────────────────────────►◄
```

statements: *PORT*
dependencies: *LAN and ATM LAN emulation connections only*
range: *X'000'–X'FFFF'*

Defines a 3-digit or 4-digit hexadecimal device number used for the attached LAN. The device number should be within the address range defined to the operating system.

The address must not be enclosed in quotation marks or apostrophes. For example, a channel device address of X'080' should be coded CUADDR=080.

This operand is required for LAN and ATM LAN emulation connections.

DELAY

```
┌─DELAY=0──────────────┐
►►─┤                    ├──────────────────────────────────────────────►◄
└─DELAY=timeout_value─┘
```

statements: *PORT*
dependencies: *LAN and ATM LAN emulation connections only (subarea node)*
range: *0–9.999*

Specifies the maximum amount of time, in seconds and thousandths of seconds, that VTAM should wait before transmitting low-priority data (that is, PIU transmission priority 0 or 1) to the physical unit.

For more efficient operation, code this value as high as possible. For maximum speed across the interface, do not code delay. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information.

DIAL

```
┌─DIAL=NO──────────┐
►►─┤                  ├──────────────────────────────────────────────►◄
│          ┌─NO─┐   │
└─DIAL=─┤            ├─┘
└─YES─┘
```

statements: *GROUP*
dependencies: *none*

Specifies whether the lines under the GROUP statement require switched line control protocols.

Note: You must code DIAL=YES for peripheral nodes and DIAL=YES when MEDIUM=HPRIP on PORT for Enterprise Extender.

DIAL=NO
Specifies that the lines in the group do not require switched line control protocols.

DIAL=YES
Specifies that the lines in the group require switched line control protocols.

DLCADDR

```
►►─┬──────────────────────────────────────────────────────────────────┬──►◄
└─DLCADDR=(─subfield_id,type,─string──)─┘
```

statements: *GROUP*
dependencies: *ATM native connections only (TGs to connection networks), DIAL=YES*

Specifies the signaling information used to establish a switched connection with a remote node through an ATM native connection network.

Multiple DLCADDR operands can be coded on a single GROUP definition statement. For example, the following can be coded on a single GROUP statement.

```
GROUP  DLCADDR=(1,TYPE1,STRING1),
       DLCADDR=(2,TYPE2,STRING2)
```

subfield_id

Indicates the subfield. A value between 1–96 can be specified.

The first DLCADDR operand must have a *subfield_id* of 1. Also, each operand must have a unique *subfield_id*. If two DLCADDR operands—following a DLCADDR operand with a subfield of 1—have the same *subfield_id*, VTAM treats the two operands as duplicates. VTAM ignores the second operand and issues an error message.

Note: If you receive an error message referring to the first DLCADDR operand coded on a GROUP statement, limited syntax checking is done on the subsequent DLCADDR operands coded.

type

Indicates the data format of the signaling information and is one of the following:

Type	Data Format
A	Compressed alphanumeric
BCD	Binary coded decimal
C	EBCDIC
D	Decimal
X	Hexadecimal

Type C (EBCDIC) is the default.

string

Defines the signaling information in the data format defined by *type*. Multiple *string* values can be coded for a *subfield_id type* pair. For example:

```
DLCADDR=(1,TYPE1,STRING1,STRING2,STRING3)
```

You can specify up to 250 bytes of signaling information on a single DLCADDR operand. If coding multiple *strings* on a single operand, the total number of bytes available for signaling information is reduced by 2 for each additional DLCADDR *string*.

The first DLCADDR operand defined for each GROUP definition statement identifies the DLC type, which for ATM native connection-network connections is ATMSVC. The remaining DLCADDR operands define the DLC signaling information for the ATM native connection-network connection. You can code only one set of DLCADDR operands on the GROUP definition statement for an ATM native connection-network connection.

Code the following DLCADDR operands for an ATM native connection-network connection. Fields in italics are user-defined fields.

```
DLCADDR=(1,C,ATMSVC,vnname,tg_use)
DLCADDR=(7,BCD,format,best_effort_indicator,a,b,c,
        forward_traffic_management_options)
DLCADDR=(8,X,format,outbound_QOS_class)
DLCADDR=(9,BCD,tns_identifier)
DLCADDR=(21,BCD,type,ATM_address)
DLCADDR=(51,X,format,a,b,c,d,e)
DLCADDR=(61,BCD,format,05,a,b)
```

The following explains the information provided by each subfield:

Subfield

Contents

1 `C` indicates EBCDIC format. `ATMSVC` indicates that the connection provides native access to an ATM network.

Code *vnname* to specify a 1–17 character network-qualified CP name for the connection network. *vnname* is reported to the network topology as a virtual node and is treated as an adjacent CP to this node. If the name is not network-qualified, the network ID of the SSCP that owns the connection network (network ID of the host) is used. For sessions that are routed through the connection network, a dynamic PU is created to represent the destination node. In effect, DYNPU=YES is enforced automatically when a session is established through the connection network.

Code *tg_use* to specify one of the following:

EXCLUSIVE
Indicates that SVCs and TGs can be established that will handle sessions only over connection network connections. Sessions through a connection network cannot be routed over an SVC already established to the same remote node if that SVC was not established through a connection network. Shared sessions can be shared between connection network connections and non-connection networks over an SVC.

SHARE
Indicates that HPR sessions routed through connection networks can use an SVC that is already established to the same remote node, even if that SVC was not established through a connection network.

7 `BCD` indicates binary coded decimal format.

Code *format* to specify one of the following:
00 Indicates ITU-T defined
03 Indicates ATM network defined

Code *best_effort_indicator* to specify one of the following:
00 Indicates that guaranteed bandwidth is desired
01 Indicates that best effort is acceptable

Code *a*, *b*, and *c* as 4-byte decimal values to specify the following:

a Forward peak cell rate (cells per second) for cell loss priority=0+1 (CLP=0+1)

b Forward sustainable cell rate (cells per second) for (CLP=0)

c Forward maximum burst size (cells) for (CLP=0)

Code *forward_traffic_management_options* to specify one of the following:

00 Indicates that tagging is not requested in the forward direction

01 Indicates that tagging is requested in the forward direction

Note: If *best_effort_indicator* is 00, code *format*, *a*, *b*, *c*, and *forward_traffic_management_options*.

If *best_effort_indicator* is 01, code *format* and *a*.

8 `X` indicates hexadecimal format.

Code *format* to specify one of the following:
00 Indicates ITU-T defined
03 Indicates ATM network defined

Code *outbound_QOS_class* to specify the QOS class for outbound (forward) data traffic.

9 `BCD` indicates binary coded decimal format.

Code *tns_identifier* to specify the carrier identification code for the carrier that will be used for this connection.

21 `BCD` indicates binary coded decimal format.

Code *type* to specify the following:

0101 Indicates public E164 address, which means the address is in a public ATM network. The address is variable in length.

Code *ATM_address* only if remote nodes that use the same connection network are attached directly to public networks. Code *ATM_address* to specify the public gateway E164 address through which this port can be reached.

51 `X` indicates hexadecimal format.

Code *format* to specify the following:
00 Indicates ITU-T defined

Code *a*, *b*, *c*, *d*, and *e* to specify the following bearer capabilities:

a Bearer class.

a can be one of the following:
01 Indicates class A
03 Indicates class C
10 Indicates class X

b Susceptibility to clipping.

b can be one of the following:
00 Not susceptible
01 Susceptible

c User plane connection configuration.

c must be:
00 Point-to-point

d If bearer class is 10, indicates the traffic type.

d can be one of the following:
00 No indication
02 Variable bit rate

e If bearer class is 10, indicates timing requirements.

e can be one of the following:
00 No indication
01 End-to-end timing required

61 `BCD` indicates binary coded decimal format.

Code *format* to specify the following:
00 Indicates ITU-T defined

Code 05 to indicate ATM adaptation layer (AAL) type 5.

Code *a* and *b* as 4-byte decimal values to specify the following AAL parameters:
a Forward CPCS-SDU maximum size
b Backward CPCS-SDU maximum size

Note: If an application using an ATM reserved bandwidth SVC connection sends RUs smaller than the forward CPCS-SDU size, the throughput for the session may be reduced. For example, if the forward CPCS-SDU size was defined or allowed to default to 8448 (Maximum RU size 8K + 256), and the data being transmitted over this ATM connection is in 1K RUs, the throughput (as defined by the connection's forward cell rate) is greatly reduced. If this ATM SVC connection is used for 1K RU data transmission, the forward CPCS-SDU size should be set to 1280 bytes (Maximum RU size of 1K + 256).

DYNADJCP

```
                  ┌─DYNADJCP=value_of_dynadjcp_start_option──────┐
►►──┬──────────────┼──────────────────────────────────────────────┼──────────────────────────────►◄
    │              │              ┌─value_of_dynadjcp_start_option─┐
    │              └─DYNADJCP=────┼────────────────────────────────┼──┘
    │                             ├─NO─────────────────────────────┤
    │                             └─YES────────────────────────────┘
    │  ┌─DYNADJCP=YES─┐  (1)
    └──┼──────────────┼──────────────────────────────────────────────┘
       └─DYNADJCP=YES─┘
```

Notes:

1 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*ATM native connections only (TGs over PVCs), DIAL=NO; ignored if CPNAME coded*

Specifies whether an ADJCP can be created dynamically for an adjacent node.

Note: This operand is ignored if CPNAME is coded on the physical unit.

DYNADJCP=NO
Indicates that the ADJCP cannot be created dynamically for an adjacent node.

DYNADJCP=YES
Indicates that the ADJCP can be created dynamically for an adjacent node.

DYNLU

```
      ┌─DYNLU=value_of_dynlu_start_option──────┐
►►────┼────────────────────────────────────────┼──────────────────────────────────►◄
      │                ┌─value_of_dynlu_start_option─┐
      └─DYNLU=─────────┼─────────────────────────────┼┘
                       ├─NO──────────────────────────┤
                       └─YES─────────────────────────┘
      ┌─DYNLU=YES─┐  (1)
      │           │
      └─DYNLU=YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*ATM native connections only (TGs over PVCs), DIAL=NO, CDRDYN*

Specifies whether dynamic allocation of CDRSC definitions is supported. If you do not code this operand, the default is the value you coded on the DYNLU start option.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, a resource that was dynamically defined as a CDRSC cannot use that PU to connect to a session.

DYNLU=NO
Specifies that dynamic allocation of CDRSC definitions is not supported. If you code DYNLU=NO and plan to connect to other nodes using CP-CP sessions, then predefine each node in the CDRSC major node. If you do not, CP-CP session setup fails.

DYNLU=YES
Specifies that dynamic allocation of CDRSC definitions is supported.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

Note: If you have coded a DYNLU value in the adjacent CP major node for the adjacent CP to which this PU is linked, that value overrides the DYNLU value you code on this PU definition statement. For additional DYNLU considerations, see “DYNLU” on page 21 under the adjacent control point major node.

DYNPU

```
      ┌─DYNPU=NO──────┐
►►────┼───────────────┼──────────────────────────────────────────────────────────►◄
      │          ┌─NO─┐
      └─DYNPU=───┼────┼┘
                 └─YES┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	GROUP (peripheral node)	*statements:*	GROUP (TGs over SVCs, TGs to connection networks)
dependencies:	None	*dependencies:*	DIAL=YES

Specifies whether a PU is to be dynamically allocated when the calling PU cannot be identified during a switched call-in operation. DYNPU applies to APPN and subarea PUs.

A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:
- MAXOUT=8
- ANS=CONT
- DISC=(YES,F)
- DYNADJCP=YES
- CPCP=YES

DYNPU=NO
Specifies that dynamic PUs are not to be used.

DYNPU=YES
Specifies that dynamic PUs can be used.

DYNPUPFX

```
►►─┬─DYNPUPFX=CN──────────┬──────────────────────────────────────────►◄
   │                      │
   └─DYNPUPFX=string──────┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	GROUP (peripheral node)	*statements:*	GROUP (TGs over SVCs, TGs to connection networks)
dependencies:	DYNPU=YES	*dependencies:*	DIAL=YES, DYNPU=YES

Specifies the first two characters of the PU name assigned when a dynamically generated PU is created for the link station. VTAM concatenates the characters specified to form a VTAM-generated character sequence.

Note: This operand is valid only if DYNPU is specified as YES.

DYNPUPFX=CN
Indicates that CN is the first two characters of the name assigned to the dynamically created PU.

Note: VTAM will add a third character “V” for a dynamic PU created for a connection network.

DYNPUPFX=*string*
Allows you to specify the first two characters of the name assigned to the dynamically created PU. If you create your own identifier instead of using the default, you must follow VTAM naming conventions. See “Restrictions on Names” on page 14 for naming conventions.

HPR

```
►►─┬─HPR=YES─┬──────────────────────────────────────────────────────►◄
   │         │
   └─HPR=YES─┘
```

statements: *PU*
dependencies: *Native ATM connections only (TGs over PVCs), DIAL=NO*

Specifies that high performance routing (HPR) support is enabled. HPR=YES is the only valid value for native ATM connections and is the default value.

IPPORT

```
►►─┬─IPPORT=12000──┬──────────────────────────────────────────────────►◄
   │               │
   └─IPPORT=number─┘
```

statements: *PORT*
dependencies: *MEDIUM=HPRIP*
range: *0-65531*

Specifies the first number of a five consecutive number range describing ports for transmitting low, medium, high, network and signal priority data.

IPTOS

```
►►─┬─IPTOS=(20, 40, 80, C0,C0)──────────────────────────────────────►◄
   │
   │
   └─IPTOS=(─┬─20──┬─,─┬─40──┬─,─┬─80──┬─,─┬─C0─────┬─,─┬─C0─────┬─)─┘
             └─low─┘   └─med─┘   └─high─┘   └─network─┘   └─signal─┘
```

statements: *PORT*
dependencies: *MEDIUM=HPRIP*
range: *00–FF*

Specifies the type of service use for TCP/IP communication. Default values will be used for each type of service value which is not entered.

The first value (20) is represented as *low*, the second value (40) as *medium*, the third value (80) as *high*, the fourth value (C0) as *network*, and the last value (C0) as *signal* for type of service.

Note: TOS values need to be consistent throughout the network.

For more information on Type of Service and Class of Service, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

ISTATUS

```
►►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────►◄
   │                         │
   └─ISTATUS=─┬─ACTIVE───┬───┘
              └─INACTIVE─┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*none*

Specifies whether a line or PU is to be activated after the first activation of the external communication adapter major node.

ISTATUS=ACTIVE
Specifies that the line or PU becomes active when its major node becomes active.

ISTATUS=INACTIVE
Specifies that the line or PU remains inactive until the operator activates it.

The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ACT command.

LIMRES

```
      ┌─LIMRES=NO──────────┐
►►────┼                    │──────────────────────────────────────►◄
      │          ┌─NO──┐   │
      └─LIMRES=──┼     │───┘
                 └─YES─┘
      ┌─LIMRES=NO─┐   (1)
      │           │
      └─LIMRES=NO─┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

For LAN and ATM LAN Emulation Connections:		**For ATM Native Connections:**	
statements:	GROUP, LINE (peripheral node)	*statements:*	GROUP, LINE (TGs over SVCs, TGs to connection networks)
dependencies:	LIMQSINT	*dependencies:*	LIMQSINT
	LU 6.2 only		LU 6.2 only
	PUTYPE=1 or 2		DIAL=YES (TGs over SVCs, TGs to connection networks)

Specifies whether a line or group of lines is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes any sessions that traverse the resource to be deactivated if no conversation

is active. A session also can be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

If you do not code a value for LIMQSINT, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

You can also use LIMRES in conjunction with the LIMINTCP start option to specify the interval to retain a free session using a limited resource for some network management flows (excluding flows using the APPN control point sessions or dependent LU server sessions). If you do not code a value for the LIMINTCP start option, resources are not considered to be limited resources, and the sessions will remain active even after all conversations have ended.

Note: This is not a meaningful operand for Enterprise Extender.

LIMRES=NO
Specifies that this line or group of lines is not to be treated as a limited resource.

LIMRES=YES
Specifies that this line or group of lines is to be treated as a limited resource. LIMRES=YES must be coded for the LIMINTCP start option to be effective.

LIVTIME

```
►►─┬─LIVTIME=10──────┬──────────────────────────────────────────►◄
   │                 │
   └─LIVTIME=time────┘
```

statements:	*PORT*
dependencies:	*MEDIUM=HPRIP*
range:	*5-25*

Specifies the Enterprise Extender logical link control liveness timer interval in seconds.

There are two timers that can affect Enterprise Extender connections, the Logical Data Link Control (LDLC) timer and the disconnect timer. The new LDLC timer controlling keywords specified on the PORT macro are LIVTIME, SRQTIME and SRQRETRY. The disconnect timer is controlled through the DISCNT keyword on the PU macro.

The LDLC timer provides LDLC with a mechanism to wait for an interval after which LDLC may consider an Enterprise Extender connection inoperative. The disconnect timer works independently of the LDLC timer, and determines the time interval during which Enterprise Extender monitors for data transfer before disconnecting the link (if no data transfer occurs).

The following describes when an Enterprise Extender link may be disconnected:

- The interval determined by LIVTIME + (SRQTIME*SRQRETRY) expires if required signal responses are not received
- The DELAY value specified by DISCNT expires if no data transfer occurs

Note: The DISCNT must be enabled by system definition if desired (refer to the DISCNT operand in the Switched Major Nodes section). In addition, if using DISCNT, the DELAY value in effect will be the minimum of the value on the dial-out PU and the dial-in PU. In order to accurately specify the DELAY value on the dial-in PU, you must be able to control which dial-in PU being is selected. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide.*

MACADDR

```
►►──MACADDR=macaddr──────────────────────────────────────────────────►◄
```

statements:	*PU*
dependencies:	*LAN and ATM LAN emulation connections only (subarea node)*

Specifies the 12-digit hexadecimal medium access control (MAC) address for a peer processor on the local area network that this PU definition statement defines.

This operand is required for LAN and ATM LAN emulation connections. X'000000000000' is not a valid MACADDR value.

MEDIUM

For LAN and ATM LAN Emulation, Connections:

```
►►MEDIUM=──┬─ATM────┬──►◄
           ├─BOXMGR─┤
           ├─CSMACD─┤
           ├─FDDI───┤
           └─RING───┘
```

statements:	PORT
dependencies:	ADAPNO, CUADDR

For ATM Native Connections:

```
►►──┬─MEDIUM=ATM─┬──►◄
    └─MEDIUM=ATM─┘
```

statements:	PORT
dependency:	PORTNAME

For Enterprise Extender Connections:

```
►►──MEDIUM=HPRIP──────►◄
```

statements:	PORT
dependency:	DIAL=YES

Identifies the type of shared access transport facility (SATF) being defined.

For LAN and ATM LAN emulation connections, this operand is required and the valid values are RING, CSMACD, FDDI, and BOXMGR.

For Enterprise Extender connections, the only valid option is HPRIP.

For ATM native connections, ATM is the only valid value and is the default value.

MEDIUM=ATM
Defines a connection that provides native access to an ATM network through an IBM S/390 Open Systems Adapter.

MEDIUM=BOXMGR
Defines a box manager. A box manager allows an IBM 3172 Nways Interconnect Controller or an IBM S/390 Open Systems Adapter to have minimal network management. It also allows generic alerts to flow from an IBM 3172 Nways Interconnect Controller or an IBM S/390 Open Systems Adapter to the NetView program.

The IBM 3172 Nways interconnect controller tries to maintain continuous availability of the SSCP-PU session when there is an external communication adapter major node configured for network management (MEDIUM=BOXMGR). The BOXMGR session ends only when either VTAM terminates or the last external communication major node associated with the IBM 3172 Nways interconnect controller comes down. If the connection is terminated for any other reason, the IBM 3172 Nways interconnect controller attempts to reestablish the connection.

MEDIUM=BOXMGR sets MEDIUM=RING and ADAPNO=255 on the PORT definition statement. It also sets DIAL=YES on the GROUP definition statement and sets CUADDR to the value you specify on this PORT definition statement.

When MEDIUM=BOXMGR, only one GROUP, LINE, and PU definition statement is allowed. Also, the only operands that can be coded for this value are ISTATUS and SPAN. ISTATUS and SPAN are optional.

The MACADDR value for this option is specified on the DIALNO operand of the PATH definition statement of the switched major node.

Note: Because there are no LUs in the external communication adapter major node, there should be no LU definition statements in the switched major node. If there are LU definition statements under the PATH definition statement, errors occur when you attempt to activate the LUs.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of coding a box manager definition.

MEDIUM=CSMACD
Defines a connection to a CSMA/CD 802.3 LAN

MEDIUM=FDDI
Defines a connection to a FDDI LAN

MEDIUM=HPRIP
Defines a connection that provides access to resources located through an IP network through Enterprise Extender.

Note: On the PORT statement, only one XCA major node with MEDIUM=HPRIP may be active at any time.

MEDIUM=RING
Defines a connection to a token-ring network

name

```
►►──name──────────────────────────────────────────────────────────►◄
```

statements:	*VBUILD, PORT, GROUP, LINE, PU*
dependencies:	*none*

On VBUILD: *name* provides the required and unique name for this major node.

On PORT: *name* provides the required name associated with a particular adapter in an IBM 3172 Nways interconnect controller with an OSA card or virtual representation of an Enterprise Extender connection.

On GROUP: *name* provides the required minor node name of the line group represented by this definition statement.

On LINE: *name* provides the required minor node name of the link represented by this definition statement.

On PU: *name* provides the required minor node name of the physical unit represented by this definition statement.

MULTGSUP

```
►►─┬─MULTGSUP=YES──────┬──────────────────────────────────────────►◄
   │                   │
   └─MULTGSUP=─┬─YES─┬─┘
               └─NO──┘
```

statements: *PU (subarea node)*
dependencies: *none*

Specifies whether Multi-Link TGs are supported on this connection. MULTGSUP=YES are for most configurations.

MULTGSUP=YES
Specifies the XIDs received from the connecting node must have the XID2 Multiple-link TG support bit turned ON, meaning Multiple-link TGs are supported. In addition, VR TG sequence number checking on inbound PIUs will be performed. MULTGSUP=YES is the default.

MULTGSUP=NO
Specifies that the XIDs received from the connecting node will be allowed to have the XID2 Multiple-link TG support bit OFF, meaning Multiple-link TGs are not supported. In addition, VR TG sequence number checking on inbound PIUs will not be performed.

Note: If not coded correctly, Subarea XCA connections could fail at activation. This parameter is introduced to support Subarea XCA connections which traverse mediums that do not support Multi_link TGs. The specific configuration which required this parameter follows:

```
VTAM A/OSA ===tkr=== 2210 ---X.25 PSDN--- NPSI/NCP === VTAM B
```

In this case, the NPSI/NCP X.25 connection does not support multi-link TGs.

NATIVE

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─NATIVE=─┬─NO──┬─┘
              └─YES─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO, BN=YES*

Specifies whether this link station represents a connection to a native node.

If NATIVE is not specified on the ADJCP statement or on the PU definition statement, the two nodes negotiate their subnet affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

No subnet negotiations will occur if NATIVE has been specified; in this case the value between the two nodes must match for connection to occur.

If you code NATIVE on both the ADJCP and PU definition statements, the values must match.

For transmission groups (TGs) between nodes, the value specified or negotiated for NATIVE must match. If a TG is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the TG attempting activation is not allowed to become active. If the activation is for a virtual-route-based TG, then the SSCP-SSCP session will also fail.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └NETID=network_id──┘
```

statements:	*PU*
dependencies:	*ATM native connections only (TGs over PVCs), DIAL=NO, XNETALS start option*

Specifies a 1–8 character network identifier. This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See “XNETALS” on page 657 for information about XNETALS.

Note: Use of the national character # is discouraged because this character might not be on keyboards produced worldwide. Because # might not be on all keyboards, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

NETID=*network_id*
If you specify NETID on the PU definition statement, VTAM ensures that when the physical unit is active, the connecting resource is within the network specified by NETID. The network ID specified by NETID is the CP NETID of the NETID where the PU resides. If you omit NETID, VTAM dynamically acknowledges the network ID during connection establishment and this network ID becomes the CP NETID. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM’s network ID.

NETID=(,NOXNETLS)
Specifies that VTAM does not allow nonnative network attachment for LU-LU sessions for this PU. When NETID=(,NOXNETLS), VTAM indicates to the NCP that this PU will use VTAM’s native network ID.

LU-LU sessions might fail if the network identifier sent in the bind from the LU is different than the network identifier specified in VTAM, even though PU-PU

connections can occur. If no network identifier is included in the bind, or if the network identifier is the same as that specified by VTAM, the LU-LU session should activate.

Note: NETID=(,NOXNETLS) is valid only when YES is specified on the XNETALS start option. If NETID=(,NOXNETLS) is coded and XNETALS=YES is not coded, NETID defaults to the native network ID, and a message is issued that NETID has the wrong value.

NN

```
►►─┬──────────────────┬──────────────────────────────────────────►◄
   └─NN=──┬─NO──┬─────┘
          └─YES─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies whether the adjacent node is expected to be a network node.

If you specify a value for NN, it is validated when a connection is attempted to an adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If NN is not specified, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

PDELAY

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:

```
                        (1)
   ┌─PDELAY=NEGLIGIB──────┐
   ├─PDELAY=TERRESTR──────┤
►►─┤                      ├──────────────────────────────────►◄
   │            ┌─TERRESTR─┐
   └─PDELAY=────┤          ├
                ├─LONG─────┤
                ├─NEGLIGIB─┤
                └─PACKET───┘
```

For ATM Native Connections:

```
   ┌─PDELAY=NEGLIGIB──────┐
►►─┤                      ├──────────────────────────────────►◄
   │            ┌─NEGLIGIB─┐
   └─PDELAY=────┤          ├
                ├─LONG─────┤
                ├─PACKET───┤
                └─TERRESTR─┘
```

Notes:

1 For connections to virtual nodes, the default value is NEGLIGIB.

statements: PORT

statements: GROUP (TGs to connection networks)

PU (TGs over PVCs)

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)

Specifies the maximum propagation delay of the link for the TG. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PORTNAME

```
►►──PORTNAME=(port_name,link_num)──────────────────────────────────────►◄
```

statements:	*PORT*
dependencies:	*ATM native connections only, CUADDR, ADAPNO*

Specifies the name of a port on an IBM S/390 Open Systems Adapter through which an ATM native connection is made.

The maximum length for PORTNAME is 8 characters.

PORTNAME is required for ATM native connections and must match a value specified on a PORTNAME operand on a TRLE definition statement in the transport resource list (TRL) major node. Both values must match the name of the ATM OSA-2 physical port as it is defined during IBM S/390 Open Systems Adapter configuration on the ATM Native Support Settings panel, which is part of the OSA Configuration pull down option accessed through the OSA/SF OS/2® interface.

Note: PORTNAME is mutually exclusive with the CUADDR and ADAPNO operands for any single PORT definition statement.

PUTYPE

For LAN and ATM LAN Emulation Connections:

```
►►─┬─PUTYPE=5──────────┬──────────────────────────────►◄
   │                   │
   └─PUTYPE=─┬─5─┬─────┘
             └─4─┘
```

For ATM Native Connections:

```
►►─┬─PUTYPE=2─┬──────────────────────────────────────►◄
```

statements:	GROUP, LINE, PU (subarea node)	*statements:*	PU
dependency:	USER=VCNS	*dependency:*	TGs over PVCs, DIAL=NO

LAN and ATM LAN Emulation Connections: Specifies the PU type.

Notes:

1. If USER=VCNS is specified on the LINE definition statement, the PUTYPE operand is not valid.
2. If you have a virtual-route-based transmission group connection established between two hosts and HPR=YES is coded on the CDRM statements for the hosts, HPR may be used over this connection.

PUTYPE=4
Specifies a connection to an IBM 37xx Communication Controller running NCP/Token-Ring interconnection (NTRI) or a connection through a CSMA/CD 802.3 LAN to a 37xx Communication Controller running NCP. Specify PUTYPE=4 only if MEDIUM=RING or MEDIUM=CSMACD on the PORT definition statement.

PUTYPE=5
Specifies a connection to a host

ATM Native Connections: Specifies the PU type.

Note: If USER=VCNS is specified on the LINE definition statement, the PUTYPE operand is not valid.

PUTYPE=2
Specifies a connection to a remote ATM PU through an IBM S/390 Open Systems Adapter. PUTYPE=2 is the only valid value for ATM native connections and is the default value.

PVCNAME

```
►►─┬─PVCNAME=pvc_name─┬──────────────────────────────►◄
```

statements:	*LINE*
dependencies:	*ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies the name of a permanent virtual channel (PVC) through which a remote node is connected.

PVCNAME is required for ATM native connections and must match the name of the PVC as it is defined during IBM S/390 Open Systems Adapter configuration in the ATM Native Settings notebook for the ATM OSA-2 physical port. The ATM Native Settings notebook is part of the OSA/SF OS/2 interface.

SAPADDR

```
►►─┬─SAPADDR=4─────────┬──────────────────────────────────────────►◄
   │                   │
   └─SAPADDR=address───┘
```

Connection Type:	LAN and ATM LAN Emulation	ATM Native	Enterprise Extender (HPRIP)
statements:	PORT	PORT	PORT
	PU (subarea node)	PU (TGs over PVCs)	PU
dependency:	USER=SNA (LINE statement)	DIAL=NO (TGs over PVCs)	
range:	4–252	4–252	4–252

On PORT: Specifies the service access point (SAP) address for the connection to a LAN, ATM or TCP/IP network through the port that this major node defines.

Note: Enterprise Extender is a connection identifier (there are no CUADDR and ADAPNO operands).

The SAP values for Enterprise Extender are multiples of 2 and for all other mediums a multiple of 4 applies. A null SAPADDR value has the same meaning as not coding the SAPADDR operand.

Code the SAPADDR operand, in addition to the CUADDR and ADAPNO operands, to route information between the LAN and VTAM.

On PU:

Specifies the service access point (SAP) address for the remote node that this PU definition statement defines.

Note: This is not applicable for MEDIUM=HPRIP.

For Enterprise Extender, there must be different SAP values for host connections. In other words, if two hosts are issuing dials, both in and out, they must be using different SAP values to represent the connection. For example, if host 1 issues a dial to host 2 with

```
local SAP = 04
remote SAP = 08
```

and the connection comes active, host 2 cannot issue a dial back to host 1 with

```
local SAP = 08
remote SAP = 04
```

because the connection represented by the SAP pair 0408 is already active. If host 2 issues the dial with the above information, the dial request will be failed and the operator will receive a sense code of **FFC80004**.

Note: Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on SAP pairs.

SECURITY

```
   ┌─SECURITY=UNSECURE──────┐
►►─┼────────────────────────┼──────────────────────────────────────────────►◄
   │                        │
   │            ┌─UNSECURE──┐
   └─SECURITY=──┼───────────┼──┘
                │           │
                ├─PUBLIC────┤
                ├─UNDERGRO──┤
                ├─SECURE────┤
                ├─GUARDED───┤
                ├─ENCRYPT───┤
                └─SHIELDED──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SPAN

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └─SPAN=span_name───┘
```

statements: *GROUP, LINE, PU*

dependencies: *NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, see the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SRQRETRY

```
►►─┬─SRQRETRY=3──────────────────┬──────────────────────────────────────►◄
   │                             │
   └─SRQRETRY=number_of_retries──┘
```

statements: *PORT*
dependencies: *MEDIUM=HPRIP*
range: *3-9*

Specifies the number of times the short request timer is retried before the port becomes inoperative. For an explanation on using SRQRETRY externally, refer to the LIVTIME operand described earlier in this section.

SRQTIME

```
►►─┬─SRQTIME=15──┬──────────────────────────────────────────────────────►◄
   │             │
   └─SRQTIME=time┘
```

statements: *PORT*
dependencies: *MEDIUM=HPRIP*
range: *3-20*

Specifies the Enterprise Extender logical link control short request timer interval in seconds. For an explanation on using SRQTIME externally, refer to the LIVTIME operand described earlier in this section.

SUBAREA

```
►►─┬─SUBAREA=inbound_xid_value─┬──────────────────────────────────────►◄
   │                           │
   └─SUBAREA=suba_addr─────────┘
```

statements: *PU*
dependencies: *LAN and ATM LAN emulation connections only (subarea node)*
range: *1–value of the MXSUBNUM start option*

Specifies the subarea address assigned to the adjacent domain's NCP or VTAM, whichever is represented by this PU definition statement. If you code SUBAREA, VTAM compares the subarea number on the XID frame with the value you code on this operand. If the values do not match, the connection ends.

The SUBAREA value must be the same as the subarea address defined for this physical unit in the adjacent domain.

SWNORDER

```
►►─┬─SWNORDER=value_of_swnorder_start_option──────────────────────────────────►◄
   │
   │                 ┌─value_of_swnorder_start_option─┐
   └─SWNORDER=───────┼─CPNAME─────────────────────────┼
                     └─STATNID────────────────────────┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		**For ATM Native Connections:**	
statements:	GROUP, LINE	*statements:*	GROUP, LINE
dependencies:	Peripheral node	*dependencies:*	TGs over SVCs TGs to connection networks DIAL=YES

Specifies the way VTAM locates a switched PU. If you do not code SWNORDER, the default is the value of the SWNORDER start option.

If VTAM does not find the PU using the SWNORDER value you specify, it searches by the station identifier. For example, if you code SWNORDER=CPNAME and VTAM cannot find a PU with a matching CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

This operand overrides the SWNORDER start option value.

SWNORDER=CPNAME
Specifies that VTAM searches for a switched PU by the CPNAME first.

SWNORDER=STATNID
Specifies that VTAM searches for a switched PU by the station identifier first.

TGN

For LAN and ATM LAN Emulation Connections:

```
►►─┬─TGN=1──────────────────────────────────────────────────────────────────►◄
   │
   │                 ┌─1──────────────────────────────┐
   └─TGN=────────────┼─ANY────────────────────────────┼
                     └─tg_number──────────────────────┘
```

statements:	PU (subarea node)
dependency:	none
range:	1–255

For ATM Native Connections:

```
►►─┬─TGN=ANY────────────────────────────────────────────────────────────────►◄
   │
   │                 ┌─ANY────────────────────────────┐
   └─TGN=────────────┼─tg_number──────────────────────┼
                     └────────────────────────────────┘
```

statements:	PU
dependency:	PUTYPE=2, CPNAME, NETID, DIAL=NO
range:	0–20

LAN and ATM LAN Emulation Connections: Specifies the TG number that represents the connection between the host to which this PU definition statement applies and either a communication controller or another host.

Coding TGN=ANY allows the adjacent subarea node to accept whatever valid TG number it receives on the XID.

You can specify TGN=ANY only for the PU at one end of a transmission group. The PU at the other end must have a valid transmission group number or default to 1. An error occurs if you specify TGN=ANY for both PUs.

ATM Native Connections: Specifies the TG number that represents the connection between this node and the destination node. This operand is valid when PUTYPE=2.

By specifying a *tg_number*, you indicate a preferred TG that might be used depending on how the partner nodes negotiate the number.

The CPNAME and NETID operands are required with the TGN operand. Note that you can specify the CPNAME operand without the TGN operand, and the TG number will be negotiated. However, if a value for TGN is specified, a value for CPNAME and NETID must also be specified.

TGN=*tg_number*
Specifies a decimal integer assigned to the transmission group.

For APPN, coding TGN=0 means that the PU does not support parallel TGs.

TGN=ANY
For subarea, allows the adjacent node to accept whatever valid transmission group number it receives on the XID. You can specify TGN=ANY for only one end of the transmission group. For the other end, specify a valid TGN number or let it default to 1. If you specify TGN=ANY for both ends, an error occurs.

For APPN, coding TGN=ANY, or not coding TGN, allows the transmission group number to be negotiated.

TGP

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └─TGP=tgp_name─┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		**For ATM Native Connections:**	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)

Specifies the name of a TG profile definition. The characteristics of the TG profile (along with any modifiers) become the characteristics of the PU. If TGP is not specified or has not been activated when the PU becomes active, default TG characteristics are assigned.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TIMER

```
     ┌─TIMER=30──────────────┐
►►───┤                       ├──────────────────────────────────────────────►◄
     └─TIMER=timeout_value───┘
```

statements: *PORT*
dependencies: *LAN and ATM LAN emulation connections only*
range: *1–255*

Specifies the number of seconds that VTAM will wait for a response after the channel is activated.

TIMER is used for control type commands. Data commands are timed by the link protocol. However, some control commands are also timed at the link level. As a result, a restriction exists the calculation of the reply timer multiplied by the number of retransmissions should be less than the value coded for TIMER.

TOPO

```
►►──┬──────────────┬──────────────────────────────────────────────────────►◄
    └─TOPO=CONNECT─┘
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Controls the way APPN connections using this link station will be reported to APPN topology and routing services for inclusion into APPN functions.

Note: Except in the switched major node, the only value for TOPO is CONNECT. If you specify APPN on the CONNTYPE start option, it is not necessary to code TOPO, as you automatically get a value of CONNECT.

TOPO=CONNECT
APPN topology and routing services receives information about the connection and its characteristics when the connection is activated. When the connection is deactivated, APPN topology and routing services receives notification that the TG is inactive.

TYPE

```
►►──TYPE=XCA──────────────────────────────────────────────────────────────►◄
```

statements: *VBUILD*

dependencies: *none*

Indicates an external communication adapter major node. This operand is required.

UPARM1

```
►►─┬─UPARM1=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM1=user-defined_operand_value──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	0–255	*range:*	0–255

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
►►─┬─UPARM2=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM2=user-defined_operand_value──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	0–255	*range:*	0–255

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM3=user-defined_operand_value──┘
```

For LAN, ATM LAN Emulation, or Enterprise Extender Connections:		For ATM Native Connections:	
statements:	PORT	*statements:*	GROUP (TGs to connection networks) PU (TGs over PVCs)
dependency:	TGs to connection networks	*dependency:*	DIAL=YES (TGs to connection networks) DIAL=NO (TGs over PVCs)
range:	0–255	*range:*	0–255

Specifies a user-defined operand value.

USER

```
►►─┬─USER=SNA──────────┬──────────────────────────────────────────►◄
   │                   │
   └─USER=─┬─SNA─┬─────┘
           └─VCNS┘
```

For LAN and ATM LAN Emulation Connections:		For ATM Native Connections:	
statements:	LINE	*statements:*	GROUP, LINE
dependencies:	Subarea node	*dependencies:*	TGs over PVCs DIAL=NO

Identifies the owner of the line. This operand is valid only when DIAL=NO is coded on the GROUP definition statement. This operand has two acceptable values:

USER=SNA
Specifies that the line uses SNA protocols to access the LAN.

Note: The MACADDR and SAPADDR operands on the accompanying PU definition statement provide the remote node's address.

USER=VCNS
Specifies that VCNS uses the line to access the LAN.

When USER=VCNS, code only one LINE definition statement for each XCA major node.

Notes:

1. If you code USER=VCNS on more than one definition statement, a warning message is issued and the value of USER defaults to SNA.
2. If you code USER=VCNS, do not code a PU definition statement for that LINE definition statement.

VERALSID

```
      ┌─VERALSID=NO──────┐
►►────┼                  ├──────────────────────────────────────────►◄
      │                  │
      │          ┌─NO─┐  │
      └─VERALSID=┼    ┼──┘
                 └─YES┘
      ┌─VERALSID=NO─┐  (1)
      │             │
      └─VERALSID=NO─┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies whether the adjacent link station name should be used to determine the link station. When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to XID3s. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

VNGROUP

```
                          (1)
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─VNGROUP=group_name───┘
```

Notes:

1 Code VNGROUP if you code VNNAME.

statements:	*PORT*
dependencies:	*LAN, ATM LAN emulation and Enterprise Extender connections only; VNNAME; DIAL=YES on GROUP statement*

Specifies the name of the GROUP containing dial-out links available for use on the connection network named on the VNNAME operand.

For sessions that are routed through the connection network, a dynamic PU is created to represent the destination node. In effect, DYNPU=YES is enforced automatically when a session is established through the connection network. The name specified on the VNGROUP operand provides the name of the group for the dynamic PU that is allocated.

VNNAME

```
                                    (1)
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─VNNAME=connection_network_cpname─┘
```

Notes:

1 Code VNNAME if you code VNGROUP.

statements: *PORT*
dependencies: *LAN, ATM LAN emulation, or Enterprise Extender connections only; VNGROUP; DIAL=YES on GROUP statement*

Specifies a 1–17 character network-qualified CPNAME for the connection network. VNNAME is reported to the network topology as a virtual node and is treated as an adjacent CP to this node. The NETID of the SSCP that owns the connection network (the NETID of the host) is used.

For sessions that are routed through the connection network, a dynamic PU is created to represent the destination node. In effect, DYNPU=YES is enforced automatically when a session is established through the connection network.

VTAMTOPO

```
     ┌─VTAMTOPO=REPORT──────────┐
►►───┤                          ├──────────────────────────────────────────────►◄
     │                ┌─REPORT──┐
     └─VTAMTOPO=──────┤         ├
                      ├─NOREPORT─┤
                      ├─NOLLINES─┤
                      ├─NOSWPUS──┤
                      ├─INCLUDE──┤
                      └─IGNORE───┘
```

statements: *VBUILD*
dependencies: *none*

```
►►──────────────────────────────────────────────────────────────────────────────►◄
     │                ┌─REPORT──┐
     └─VTAMTOPO=──────┤         ├
                      ├─NOREPORT─┤
                      ├─NOLLINES─┤
                      └─NOSWPUS──┘
```

statements: *GROUP*
dependencies: *(switched logical lines only)*

```
►►──────────────────────────────────────────────────────────────────────────────►◄
     │                ┌─REPORT──┐
     └─VTAMTOPO=──────┤         ├
                      └─NOREPORT─┘
```

```
►►──────────────────────────────────────────────────────────────────────────────►◄
```

statements: *LINE*
dependencies: *(switched logical lines only)*

Specifies whether the VTAM topology agent reports the major nodes or subordinate minor nodes to the CMIP manager application programs for SNA local topology.

Specifying VTM.VTAMTOPO allows VTAMTOPO to be handled appropriately by the NCP Definition Facility for versions of NDF that do not recognize VTAMTOPO natively.

VTAMTOPO=REPORT

On VBUILD: Specifies that the VTAM topology agent reports this XCA major node and all subordinate minor nodes, including connected switched PUs under switched logical lines. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent reports all subordinate lines and connected switched PUs. This does not override any value specified on the LINE.

On LINE: Specifies that the VTAM topology agent reports this logical LINE.

VTAMTOPO=NOREPORT

On VBUILD: Specifies that the VTAM topology agent does not report the switched logical lines or connected switched PUs. The topology agent still reports the major node and the other subordinate resources. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent does not report any subordinate lines and connected switched PUs. This does not override any value specified on the LINE.

On LINE: Specifies that the VTAM topology agent does not report this logical LINE.

VTAMTOPO=NOLLINES

On VBUILD: Specifies that the VTAM topology agent does not report the switched logical lines. The topology agent still reports the major node and the other subordinate resources, including the connected switched PUs. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent does not report any subordinate switched logical lines. The topology agent still reports the connected switched PUs under the subordinate switched logical lines. This does not override any value specified on the LINE.

VTAMTOPO=NOSWPUS

On VBUILD: Specifies that the VTAM topology agent does not report the connected switched PUs. The topology agent still reports the major node and the other subordinate resources, including the switched logical lines. This does not override any value specified on the GROUP.

On GROUP: Specifies that the VTAM topology agent does not report any connected switched PUs under the subordinate switched logical lines. The topology agent still reports the subordinate switched logical lines.

VTAMTOPO=INCLUDE

On VBUILD: Specifies that the VTAM topology agent reports this XCA major node and all subordinate minor nodes. VTAMTOPO=INCLUDE overrides any other value specified on the GROUP or LINE.

VTAMTOPO=IGNORE

On VBUILD: Specifies that the VTAM topology agent does not report this XCA major node nor any subordinate minor nodes. VTAMTOPO=IGNORE overrides any other value specified on the GROUP or LINE.

XID

```
   ┌─XID=YES─┐
►►─┴─────────┴──────────────────────────────────────────────────────►◄
```

statements: *PU*
dependencies: *ATM native connections only (TGs over PVCs), DIAL=NO*

Specifies that a PU type 2.1 channel contact procedure is to be used. XID=YES is the only valid value for ATM native connections and is the default value.

XNETALS

```
   ┌─XNETALS=value_of_XNETALS_start_option─┐
►►─┴───────────────────────────────────────┴──────────────────────────►◄
   └─XNETALS=─┬─NO──┬──────────────────────┘
              └─YES─┘
```

statements: *PU*
dependencies: *none*

Determines whether the network ID of an attaching node or the VTAM network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true. If the NETID that the PU sent in its XID is not the same as the VTAM ID and XNETALS=NO or is defaulted, IST680I will be issued.

XNETALS=NO
Indicates that the VTAM network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement, then it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement, then VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES
Indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Note: Providing that XNETALS=YES is coded, you can prohibit nonnative network attachment for LU-LU sessions for a particular PU by coding XNETALS=NO on the GROUP, LINE or PU statements.

Local Non-SNA Major Node

To define a local non-SNA major node, code an LBUILD definition statement followed by a LOCAL definition statement. The definition statements define each channel-attached non-SNA terminal as part of a logical set (group) of channel-attached non-SNA terminals. The non-SNA cluster controller (3272 or compatible device) to which the terminal is attached is not defined to VTAM. All terminals defined within a local non-SNA major node do not need to be on the same controller. Different terminals on the same controller can be defined to VTAM in different local non-SNA major nodes.

Table 26. Local Non-SNA Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	LBUILD	CONFGDS	194				
		CONFGPW	195				CONFGDS
name	LOCAL	CUADDR	195	R			
		TERM	201	R			
		ASLENT	193				ASLTAB
		ASLTAB	194				
		BUFNUM	194				
		DLOGMOD	195				MODETAB
		FEATUR2	196			(NOEDATS, MODEL1, NOPRINTR, NOSELPEN)	
		ISTATUS	197			ACTIVE	
		LOGAPPL	197				NetView or TSO
		LOGTAB	198				dependent LUs

Table 26. Local Non-SNA Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		MDLENT	198				MDLTAB
		MDLTAB	198				
		MODETAB	199			ISTINCLM	
		REGISTER	200			NETSRVR	APPN only
		SPAN	200				NetView
		SSCPFM	200			USS3270	
		USSTAB	201			ISTINCDT	Dependent LUs

Legend:

R Required operand.

Coding Notes

LBUILD Definition Statement

Code one LBUILD definition statement for each logical group of channel-attached non-SNA terminals.

LOCAL Definition Statement

Code one LOCAL definition statement for each channel-attached non-SNA terminal in the VTAM domain.

Full Syntax

The full syntax for the local non-SNA major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

LBUILD

```
►►──┬──────────┬──LBUILD──┬──────────────────────────────────────────────────┬──►◄
    └─name─────┘          ├─CONFGDS=confgds_name──────────────────────────────┤
                          └─CONFGPW=confgds_password,CONFGDS=confgds_name──────┘
```

LOCAL

```
►►──name──LOCAL──┬──────────────────────────────────────────┬──────────────────────►
                 └─ASLENT=associated_lu_table_entry_name────┘

►──┬──────────────────────────────────────┬──┬──────────────────────────────────┬──►
   └─ASLTAB=associated_lu_table_name──────┘  └─BUFNUM=number_IO_buffers──────────┘

►──CUADDR=device_address──┬──────────────────────────────────────────────────┬──►
                          └─DLOGMOD=default_logon_mode_table_entry_name──────┘
```

```
►─┬─FEATUR2=(MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────────►
  │
  └─FEATUR2=(─┬─MODEL1──┬─┬─NOEDATS──┬─┬─NOPRINTR──┬─┬─NOSELPEN──┬─)─┘
              └─MODEL2──┘ └─EDATS────┘ └─PRINTR────┘ └─SELPEN────┘

►─┬─ISTATUS=ACTIVE──────┬─┬─LOGAPPL=controlling_primary_lu─┬──────────────►
  │                     │ │                                │
  └─ISTATUS=─┬─ACTIVE───┬─┘ │                                │
             └─INACTIVE──┘   │                                │

►─┬─LOGTAB=interpret_table_name─┬─┬─MDLENT=model_name_table_entry_name─┬──►

►─┬─MDLTAB=model_name_table_name─┬─┬─MODETAB=ISTINCLM──────────────────┬──►
  │                              │ │                                    │
  │                              │ └─MODETAB=logon_mode_table_name──────┘

►─┬─REGISTER=NETSRVR────┬─┬─SPAN=span_name─┬─┬─SSCPFM=USS3270──────┬──────►
  │                     │ │                │ │                     │
  └─REGISTER=─┬─NETSRVR─┬─┘ │                │ └─SSCPFM=─┬─USS3270─┬─┘
              ├─CDSERVR─┤   │                │           └─USS3275─┘
              └─NO──────┘   │                │

►─TERM=─┬─3277─┬──────────────────────────────────────────────────────────►
        ├─3279─┤
        ├─3284─┤
        └─3286─┘

►─┬─USSTAB=ISTINCDT──────┬──────────────────────────────────────────────►◄
  │                      │
  └─USSTAB=uss_table_name─┘
```

Operand Descriptions

The following section describes the operands you can code for the local non-SNA major node.

ASLENT

```
►►─┬─ASLENT=associated_lu_table_entry_name─┬──────────────────────────►◄
```

statements:	*LOCAL*
dependencies:	*ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    └─ASLTAB=associated_lu_table_name──┘
```

statements: *LOCAL*
dependencies: *none*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

BUFNUM

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    └─BUFNUM=number_IO_buffers─────────┘
```

statements: *LOCAL*
range: *1–255*

Specifies the number of IO buffers to be read from devices that are not properly emulating 3x74 type devices. BUFNUM helps ensure that the screen is fully read when VTAM issues READ and READ modifies to the device.

BUFNUM should be set so that BUFNUM*IOBUF_*size* is large enough to get all of the data from a screen.

CONFGDS

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    └─CONFGDS=name─────────────────────┘
```

statements: *LBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─CONFGPW=password─────────────┘
```

statements:	*LBUILD*
dependencies:	*CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

CUADDR

```
►►──CUADDR=device_address──────────────────────────────────────────────►◄
```

statements:	*LOCAL*
dependencies:	*none*
range:	*X'000'–X'FFFF'*

Defines the 3-digit or 4-digit hexadecimal device number used for this channel-attached terminal. The address must not be enclosed in quotation marks or apostrophes. For example, a channel device address of X'080' is coded as CUADDR=080.

Note: A local non-SNA terminal should not be defined to and activated by VTAM if its device number is defined as an MVS console and allocated to console services. Activating a local non-SNA terminal whose device number is in use by console services can cause VTAM, console services, or both to abend.

DLOGMOD

```
►►──┬──────────────────────────────────────────┬──────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name┘
```

statements:	*LOCAL*
dependencies:	*MODETAB*

Specifies the name of the default logon mode table entry used by default, if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the IBM-supplied session parameters for BSC 3270 terminals.

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on logon mode entries, see “Logon Mode Table” on page 698.

FEATUR2

```
►► ┌─FEATUR2=(MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────┐
   │                                                                      │
   │                ┌─MODEL1──┐   ┌─NOEDATS─┐   ┌─NOPRINTR─┐   ┌─NOSELPEN─┐
   └─FEATUR2=(──────┼─────────┼───┼─────────┼───┼──────────┼───┼──────────┼──────)─┘
                    └─MODEL2──┘   └─EDATS───┘   └─PRINTR───┘   └─SELPEN───┘
                                                                          ►◄
```

statements: *LOCAL*
dependencies: *none*

Identifies the machine features for a channel-attached non-SNA terminal.

FEATUR2=DUALCSE

FEATUR2=LOWERCSE
Specifies how to send alphabetical characters coded with the TEXT operand on a USSMSG macroinstruction to a non-SNA terminal over the SSCP-LU session. This value does not affect non-alphabetical characters or any characters coded on the BUFFER operand of a USSMSG macroinstruction.

Code LOWERCSE to indicate that alphabetical characters are sent to the terminal over the SSCP-LU session in lowercase. Code DUALCSE to indicate that VTAM sends all characters as they are coded in the USSMSG macroinstruction.

FEATUR2=EDATS

FEATUR2=NOEDATS
Specifies whether this terminal has the extended data stream feature.

If you code EDATS, you must specify a logon mode entry that has a PSERVIC operand that indicates extended data streaming and that also contains other required device-specific and application-specific bit combinations. Refer to the device’s component description manual and the application’s programming manual for information on the specific bit setting to code.

FEATUR2=MODEL1

FEATUR2=MODEL2
Identifies the specific model number (Model 1 or 2) for this 3277, 3284, or 3286 terminal. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

These FEATUR2 values are available to an application program as part of the device characteristics pertaining to this channel-attached terminal. You can obtain these characteristics using the INQUIRE macroinstruction. For more information on the INQUIRE macroinstruction, see *OS/390 eNetwork Communications Server: SNA Programming*.

FEATUR2=NOPRINTR

FEATUR2=PRINTR
Specifies whether the local device is a printer.

If you do not define a local non-SNA printer as a printer, raising the cover on the printer could end the session between the printer and the application. By defining it as a printer, you specify to VTAM that an asynchronous device-end from this device (caused by lifting the cover) is not to end the session.

See also the description of the VTAM start option ASYDE under “ASYDE” on page 562.

FEATUR2=NOSELPEN

FEATUR2=SELPEN
Specifies whether this terminal supports the use of a selector pen.

ISTATUS

```
   ┌─ISTATUS=ACTIVE──────────┐
►►─┼─────────────────────────┼──────────────────────────────────────────────►◄
   │                ┌─ACTIVE─┐│
   └─ISTATUS=───────┼────────┼┘
                    └─INACTIVE┘
```

statements: *LOCAL*
dependencies: *none*

Determines whether this terminal is to be activated after the first activation of the local non-SNA major node. The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ACT command.

ISTATUS=ACTIVE
Specifies that the terminal becomes active when its local non-SNA major node becomes active.

ISTATUS=INACTIVE
Specifies that the terminal remains inactive until the operator activates it.

LOGAPPL

```
►►──┬──────────────────────────────┬──────────────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu┘
```

statements: *LOCAL*
dependencies: *NetView or TSO*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─LOGTAB=interpret_table_name──┘
```

statements:	*LOCAL*
dependencies:	*valid only for dependent logical units*

Specifies the name of an interpret table that VTAM uses when processing logon requests originating from the logical unit defined by this LOCAL definition statement. VTAM uses the interpret table to interpret the name received from an initiation request. See “Interpret Table” on page 690 for a description of how to define an interpret table.

MDLENT

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name┘
```

statements:	*LOCAL*
dependencies:	*MDLTAB*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the associated LU table by default.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─┘
```

statements:	*LOCAL*

dependencies: *none*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

The model name is only passed when VTAM initiates the session with the application program, for example, only when the LOGAPPL = parameter is also coded.

If you omit the LOGAPPL operand or the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information on the MODIFY TABLE command.

MODETAB

```
►►─┬─MODETAB=ISTINCLM──────────────┬──────────────────────────────────────►◄
   │                               │
   └─MODETAB=logon_mode_table_name─┘
```

statements: *LOCAL*
dependencies: *none*

Specifies the name of a logon mode table to be used for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►─┬──────┬──────────────────────────►◄   ►►─name──────────────────────────►◄
   └─name─┘
```

statements: LBUILD *statements:* LOCAL

Assigns a name to the resource being defined by the definition statement.

On LBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On LOCAL: Provides the required minor node name assigned to the device whose address appears in the CUADDR operand in this LOCAL definition statement.

REGISTER

```
►►─┬─REGISTER=NETSRVR──────────┬──────────────────────────────────────►◄
   │                           │
   └─REGISTER=─┬─NETSRVR─┬─────┘
               │         │
               ├─CDSERVR─┤
               └─NO──────┘
```

statements:	*LOCAL*
dependencies:	*APPN only*

Specifies how a resource should be registered.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

REGISTER=NO
Indicates that the end node resource should not be registered.

SPAN

```
►►─┬──────────────────────┬──────────────────────────────────────────►◄
   └─SPAN=span_name───────┘
```

statements:	*LOCAL*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SSCPFM

```
   ┌─SSCPFM=USS3270──────┐
►►─┼─────────────────────┼──────────────────────────────────────────────►◄
   │            ┌─USS3270─┐│
   └─SSCPFM=────┼─────────┼┘
                └─USS3275─┘
```

statements: *LOCAL*
dependencies: *none*

Determines what type of RUs the logical unit or terminal can support in its communications with the SSCP.

SSCPFM=USS3270
Allows a user-defined logon message (USSMSG10) to print on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

SSCPFM=USS3275
Prevents a user-defined logon message (USSMSG10) from printing on a terminal defined as an SDLC 3275 (PUTYPE=1) that might have an attached printer.

Code SSCPFM=USS3275 to prevent user-defined messages from printing on BSC printers. However, be aware that coding this value prevents the local copy hardware function key from working with certain BSC printers. To be sure this function works, code SSCPFM=USS3270 or take the default.

TERM

```
►►─TERM=──┬─3277─┬──────────────────────────────────────────────────────►◄
          ├─3279─┤
          ├─3284─┤
          └─3286─┘
```

statements: *LOCAL*
dependencies: *none*

Specifies a specific, channel-attached non-SNA terminal (printer or display station).

USSTAB

```
   ┌─USSTAB=ISTINCDT──────────┐
►►─┼──────────────────────────┼──────────────────────────────────────────►◄
   └─USSTAB=uss_table_name────┘
```

statements: *LOCAL*
dependencies: *for dependent LUs only*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

A terminal user can issue a USS command by coding the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides

the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

Local SNA Major Node

To define a local SNA major node, code a VBUILD definition statement for the major node and a separate PU and LU definition statement for each minor node.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of requirements for defining SNA and non-SNA devices.

Table 27. Local SNA Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LOCAL	236	R			
		CONFGDS	213				
		CONFGPW	213				CONFGDS
name	PU	AUTHLEN	211			YES	APPN only Type 2.1
		CAPACITY	211			8K	APPN only
		CONNTYPE	213			CONNTYPE start option	APPN only; NODETYPE
		COSTBYTE	214			0	APPN only
		COSTTIME	215			0	APPN only
		CPCP	215			CPCP start option	APPN only
		CPNAME	215				
		CUADDR	215				TRLE
		DELAY	216			0.200	
		DISCNT	216			(NO)	
		DYNADJCP	217			DYNADJCP start option	APPN only; CPNAME
		DYNLU	218			DYNLU start option	
		ENCR	219		P	NONE	
		ENCRTYPE	220		P	DES	
		HPR	221			YES	APPN only
		INCLUD0E	221			NO	
		ISTATUS	222			ACTIVE	
		LIMRES	222			NO	LIMQSINT; LU 6.2; PUTYPE=1, 2
		LUGROUP	225				
		LUSEED	225				LUGROUP
		MAXDATA	226			261 or 265	PU type 2
		MAXBFRU	226			1 or 5	TRLE
		NATIVE	229				BN=YES APPN only
		NETID	229				XNETALS

Table 27. Local SNA Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		NN	230				APPN only
		PDELAY	231			TERRESTR	APPN only
		PUTYPE	231			2	
		SECNET	233			NO	3174 token ring
		SECURITY	233			UNSECURE	APPN only
		SPAN	234				NetView
		TGN	234			ANY	APPN only
		TGP	235				APPN only
		TOPO	235			CONNECT	APPN only
		TRLE	236				APPN only CUADDR
		UPARM1	236			128	APPN only
		UPARM2	237			128	APPN only
		UPARM3	237			128	APPN only
		VERALSID	237			NO	APPN only
		XID	238			NO	
		XNETALS	239			XNETALS start option value	
name	LU	LOCADDR	223	R			see description
		ASLENT	210		P		ASLTAB
		ASLTAB	210		P		
		CERTIFY	212		P	NO	ENCR
		CKEYNAME	212		P		
		CLRSESSQ	212		P	NO	
		DLOGMOD	217		P		MODETAB
		EAS	219		P	256	independent LUs only
		ENCR	219		P	NONE	
		ENCRTYPE	220		P	DES	
		FASTPASS	220		P	YES	TSO only
		ISTATUS	222			ACTIVE	
		LOGAPPL	223		P		NetView or TSO
		LOGTAB	224		P		dependent LUs
		LUDR	224			YES	
		MDLENT	227		P		MDLTAB
		MDLTAB	227		P		
		MODETAB	228		P	ISTINCLM	
		PACING	230		P	1	
		REGISTER	232		P	see description	APPN only
		RESETLU	232			YES	dependent LUs
		SPAN	234				NetView
		SSCPFM	234		P	FSS	
		USSTAB	237		P	ISTINCDT	dependent LUs
		VPACING	238		P	1	

Table 27. Local SNA Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies

Legend:

R Required operand.

P Code this operand on the higher-level PU definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each set of channel-attached SNA devices. One VBUILD definition statement, placed before the first PU definition statement, must be included for each major node.

PU Definition Statement

Code a PU definition statement for each physical unit, such as a cluster controller, in the local SNA major node. Note that a physical unit and all its logical units must be defined within a single major node.

LU Definition Statement

Code an LU definition statement for each logical unit associated with a physical unit within a local SNA major node. The LU definition statement must follow the PU definition statement that defines the physical unit with which this logical unit is associated.

Dynamic Change of Operands

You can dynamically change the following local SNA major node operands by editing the VTAMLST and then issuing the VARY ACT,UPDATE=ALL command. You can specify the new operand value on a higher-level definition statement for sifting, if applicable.

ASLENT
ASLTAB
DLOGMOD
ENCR
ENCRTYPE
FASTPASS
ISTATUS *
LOCADDR
LOGAPPL
LOGTAB
MDLENT
MDLTAB
MODETAB
PACING
REGISTER
SSCPFM
USSTAB
VPACING

Note: * You can dynamically change ISTATUS only on the LU definition statement.

To dynamically change the operand, the resource to which it applies must be inactive. For instance, an operand on an LU definition statement which receives its value from a PU definition statement can be dynamically changed only when the LU is inactive; the PU can remain active.

You can dynamically change the operands associated with tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) at any time without inactivating resources.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and *OS/390 SecureWay Communications Server: SNA Operation* for more information on the VARY ACT,UPDATE=ALL command.

NODETYPE Start Option

Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

Full Syntax

The full syntax for the local SNA major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=LOCAL──────────────────────────────────────►
    └─name─────┘

►──┬──────────────────────────────────────────────────────────────────────►◄
   ├─CONFGDS=confgds_name──────────────────────────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────────────────────────┘
```

PU

```
►►──name──PU──┬─AUTHLEN=YES──────────┬──┬─CAPACITY=32M──────────────────┬──►
              └─AUTHLEN=──┬─NO──┬──────┘  └─CAPACITY=number_of_bits────────┘
                          └─YES─┘

►──┬──┬─CONNTYPE=value_of_conntype_start_option──────────┬──────────────────►
   │  │                                                  │
   │  └─CONNTYPE=──┬─value_of_conntype_start_option──┬──┘
   │               ├─APPN────────────────────────────┤
   │               └─LEN─────────────────────────────┘
   │  ┌─CONNTYPE=APPN──┐   (1)
   └──┤                │──────────────────────────────────────────────────
      └─CONNTYPE=APPN──┘
```

Local SNA Major Node

```
►─┬─COSTBYTE=0──────────────────┬─┬─COSTTIME=0──────────────────┬──────────────►
  │                             │ │                             │
  └─COSTBYTE=cost_per_byte──────┘ └─COSTTIME=cost_per_unit_time─┘

►─┬─CPCP=value_of_cpcp_start_option──────────┬──────────────────────────────────►
  │                                          │
  │          ┌─value_of_cpcp_start_option──┐ │  └─CPNAME=control_point_name──┘
  └─CPCP=────┼─NO──────────────────────────┼─┘
             └─YES─────────────────────────┘

►──────────────────────────┬─┬─DELAY=.2──────────────────┬──────────────────────►
  └─CUADDR=device_address──┘ │                           │
                             └─DELAY=timeout_value───────┘

►─┬─DISCNT=(NO)──────────────────────────────────────────────────────────────►
  │
  │            ┌─(NO)──,F──────────┐
  └─DISCNT=────┼─(YES──┬──────────┬──)──┘
               │       ├─,NF──────┤
               │       └──────────┘

►─┬─DYNADJCP=value_of_dynadjcp_start_option──────────┬──────────────────────────►
  │                                                  │
  │          ┌─value_of_dynadjcp_start_option──┐    │
  │  └─DYNADJCP=────┼─NO──────────────────────────┼──┘
  │             └─YES─────────────────────────┘
  │  ┌─DYNADJCP=YES──┐  (2)
  └──┼──────────────┼──────────────────────────────┘
     └─DYNADJCP=YES──┘

►─┬─DYNLU=value_of_dynlu_start_option──────────┬─┬─ENCR=NONE──────────┬──────►
  │                                            │ │                    │
  │          ┌─value_of_dynlu_start_option──┐  │ │  └─ENCR=──┬─NONE──┬──┘
  │  └─DYNLU=────┼─NO──────────────────────────┼──┘ │           ├─OPT───┤
  │             └─YES─────────────────────────┘    │           ├─REQD──┤
  │  ┌─DYNLU=YES──┐  (3)                          │           └─SEL───┘
  └──┼───────────┼──────────────────────────────────┘
     └─DYNLU=YES──┘

►─┬─ENCRTYPE=DES──────┬─┬─HPR=YES──────────┬─┬─INCLUD0E=NO──────────┬──────────►
  │                    │ │                  │ │                      │
  └─ENCRTYPE=TDES24────┘ └─HPR=──┬─NO──┬────┘ └─INCLUD0E=──┬─NO──┬────┘
                                 └─YES─┘                    └─YES─┘
```

```
►─┬─ISTATUS=ACTIVE──────────┬─┬─LIMRES=NO──────────┬──────────────────────────►
  │                         │ │                    │
  └─ISTATUS=─┬─ACTIVE───┬───┘ └─LIMRES=─┬─NO───┬───┘
             └─INACTIVE─┘               └─YES──┘
                                 ┌─LIMRES=NO──┐ (4)
                                 └─LIMRES=NO──┘

►─┬──────────────────────────────────────────────────────────────────────────►
  ├─LUGROUP=lu_group_name──────────────────────────────────────────────────┤
  └─LUGROUP=lu_group_name,LUSEED=lu_pattern_name─────────────────────────┘

            (6)                          (8)
  ┌─MAXBFRU=5──────────┐   ┌─MAXDATA=265──────────┐
            (5)                          (7)
  ├─MAXBFRU=1──────────┤   ├─MAXDATA=261──────────┤
►─┤                    ├───┤                      ├──────────────────────────►
  └─MAXBFRU=number_of_buffers─┘   └─MAXDATA=number_of_bytes─┘

►─┬─NATIVE=─┬─NO─┬─┬─NETID=network_id─┬─┬─NN=─┬─NO─┬─┬──────────────────────►
  │         └─YES┘ │                  │ │     └─YES┘ │
  └──────────────────┘                  └────────────┘

  ┌─PDELAY=TERRESTR──────┐
                (9)
  ├─PDELAY=NEGLIGIB──────┤                   ┌─SECNET=NO──────────┐
►─┤                      ├─┬─PUTYPE=2─┬──────┤                    ├──────────►
  └─PDELAY=─┬─TERRESTR─┬─┘ │          │      └─SECNET=─┬─NO───┬─┘
            ├─LONG─────┤   │          │                └─YES──┘
            ├─NEGLIGIB─┤   │          │
            └─PACKET───┘   │          │

  ┌─SECURITY=UNSECURE────────┐                          ┌─TGN=ANY──────────┐
►─┤                          ├─┬─SPAN=span_name─┬─┬──────┤                  ├──────►
  └─SECURITY=─┬─UNSECURE─┬─┘ │                │ │      └─TGN=─┬─ANY──────┬─┘
              ├─PUBLIC───┤   │                │ │             └─tg_number─┘
              ├─UNDERGRO─┤   │                │ │
              ├─SECURE───┤   │                │ │
              ├─GUARDED──┤   │                │ │
              ├─ENCRYPT──┤   │                │ │
              └─SHIELDED─┘   │                │ │

►─┬─TGP=tgp_name─┬─┬─TOPO=CONNECT─┬─┬─TRLE=─┬─trle_name─┬─┬──────────────────►
  │              │ │              │ │       └─*─────────┘ │
  └──────────────┘ └──────────────┘ └─────────────────────┘
```

Local SNA Major Node

```
►─┬─UPARM1=128──────────────────────────┬──────────────────────────────────►
  └─UPARM1=user-defined_operand_value───┘

  ┌─UPARM2=128──────────────────────────┐
►─┴─UPARM2=user-defined_operand_value───┴──────────────────────────────────►

►─┬─UPARM3=128──────────────────────────┬──┬─┬─VERALSID=NO─────────┬─┬──────►
  └─UPARM3=user-defined_operand_value───┘  │ │                     │ │
                                           │ └─VERALSID=─┬─NO────┬─┘ │
                                           │             └─YES───┘   │
                                           │ ┌─VERALSID=NO─┬─(10)  │
                                           └─┴─VERALSID=NO─┘       ┘

  ┌─XID=NO──────────┐  ┌─XNETALS=value_of_XNETALS_start_option─┐
►─┴─XID=─┬─NO────┬─┴──┴─XNETALS=─┬─NO────────────────────────┬─┴──────────►◄
         └─YES───┘               └─YES───────────────────────┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

2 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

3 DYNLU=YES is the only valid value if the TRLE operand is coded.

4 LIMRES=NO is the only valid value if the TRLE operand is coded.

5 MAXBFRU defaults to 1 when HPR=NO, XID=NO, CONNTYPE=LEN, or the VTAM node is not HPR capable.

6 MAXBFRU defaults to 5 when HPR=YES, XID=YES, CONNTYPE=APPN, and the VTAM node is HPR capable.

7 MAXDATA defaults to 261 when PUTYPE=1 is coded.

8 MAXDATA defaults to 265 when PUTYPE=2 is coded.

9 NEGLIGIB is the default if the TRLE operand is coded.

10 VERALSID=NO is the only valid value if the TRLE operand is coded.

LU

```
►►─name──LU──LOCADDR=address──┬──────────────────────────────────────────►
                              └─ASLENT=associated_lu_table_entry_name─┘
```

```
                                                  ┌─CERTIFY=NO──────────┐
►──┬──────────────────────────────────────────────┴──────────────────────┴──────────────────────►
   └─ASLTAB=associated_lu_table_name─┘            └─CERTIFY=─┬─NO──┬─┘
                                                             └─YES─┘

   ┌─CKEYNAME=name──────────────┐   ┌─CLRSESSQ=NO──────────┐
►──┴────────────────────────────┴───┴──────────────────────┴──────────────────────────────────►
   └─CKEYNAME=crypto_key_name─┘    └─CLRSESSQ=─┬─NO──┬─┘
                                               └─YES─┘

                                                                  ┌─EAS=256──────────────┐
►──┬──────────────────────────────────────────────────────────────┴──────────────────────┴──────►
   └─DLOGMOD=default_logon_mode_table_entry_name─┘                └─EAS=number_of_sessions─┘

   ┌─ENCR=NONE──────────┐   ┌─ENCRTYPE=DES──────┐   ┌─FASTPASS=YES──────────┐
►──┴─────────────────────┴───┴────────────────────┴───┴───────────────────────┴──────────────────►
   └─ENCR=─┬─NONE──┬─┘       └─ENCRTYPE=TDES24─┘       └─FASTPASS=─┬─YES──┬─┘
           ├─OPT───┤                                                └─NO───┘
           ├─REQD──┤
           └─SEL───┘

   ┌─ISTATUS=ACTIVE──────────┐
►──┴─────────────────────────┴──┬──────────────────────────────────────────────────────────────►
   └─ISTATUS=─┬─ACTIVE──┬─┘    └─LOGAPPL=controlling_primary_lu─┘
              └─INACTIVE─┘

                                                  ┌─LUDR=YES──────────┐
►──┬──────────────────────────────────────────────┴───────────────────┴──────────────────────────►
   └─LOGTAB=interpret_table_name─┘                └─LUDR=─┬─YES──┬─┘
                                                          └─NO───┘

►──┬──────────────────────────────────────────────────────────────────────────────────────────────►
   └─MDLENT=model_name_table_entry_name─┘

                                                  ┌─MODETAB=ISTINCLM──────────────────┐
►──┬──────────────────────────────────────────────┴───────────────────────────────────┴──────────►
   └─MDLTAB=model_name_table_name─┘               └─MODETAB=logon_mode_table_name─┘
```

Local SNA Major Node

```
                                                  (2)
                                   ┌REGISTER=NO──────────┐
                                                   (1)
 ┌PACING=1──────────────┐          ├REGISTER=NETSRVR────┤
►┤                      ├──────────────────────────────────────────►
 └PACING=┌1──────────┐  │          │            ┌NO────┐
         │           │  │          │            ├NETSRVR┤
         └pacing_number┘  │          └REGISTER=   └CDSERVR┘

 ┌RESETLU=YES───────┐                          ┌SSCPFM=FSS────────┐
►┤                  ├──┬──────────────────────┬──────────────────────►
 └RESETLU=┌YES┐      │  └SPAN=span_name────┘  └SSCPFM=┌FSS────┐
          ├───┤      │                                ├───────┤
          └NO─┘      │                                └USSSCS─┘

 ┌USSTAB=ISTINCDT──────┐  ┌VPACING=1──────────────┐
►┤                     ├──┤                       ├──────────────────►◄
 └USSTAB=uss_table_name┘  └VPACING=number_of_requests┘
```

Notes:

1 REGISTER defaults to NETSRVR for dependent LUs.

2 REGISTER defaults to NO for independent LUs.

Operand Descriptions

The following section describes the operands you can code for the local SNA major node.

ASLENT

```
►►──┬──────────────────────────────────────┬──────────────────────►◄
    └ASLENT=associated_lu_table_entry_name┘
```

statements: *PU, LU*
dependencies: *ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►──┬──────────────────────────────────┬──────────────────────►◄
    └ASLTAB=associated_lu_table_name┘
```

statements: *PU, LU*
dependencies: *none*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information on the MODIFY TABLE command.

AUTHLEN

```
►►─┬─AUTHLEN=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─AUTHLEN=──┬─NO──┬────┘
               └─YES─┘
```

statements:	*PU*
dependencies:	*Type 2.1; APPN only*

Specifies whether VTAM should pass the transmission priority field specified by this PU to another PU.

AUTHLEN=NO
VTAM will not pass the specified transmission priority, instead the default priority of medium will be passed.

AUTHLEN=YES
VTAM will pass the specified transmission priority.

CAPACITY

```
►►─┬─CAPACITY=32M──────────────┬──────────────────────────────────────►◄
   │                           │
   └─CAPACITY=number_of_bits───┘
```

statements:	*PU*
dependencies:	*APPN only*
range:	*1K–1000M*

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either Kb per second (for example, 100K) or Mb per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Note: Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are

close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CERTIFY

```
►►─┬─CERTIFY=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─CERTIFY=─┬─NO──┬────┘
              └─YES─┘
```

statements: *PU, LU*
dependencies: *CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner's LU support also supports the CERTIFY function.*

Specifies whether VTAM, on behalf of the application program, will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CKEYNAME

```
►►─┬─CKEYNAME=name──────────────┬──────────────────────────────────►◄
   │                            │
   └─CKEYNAME=crypto_key_name───┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of the key-encrypting key (KEK) in the cryptographic key data set (CKDS) for the LU. Use CKEYNAME to encipher the session key that is sent in the BIND to the LU when a cryptographic session is established. CKEYNAME has the same format and restrictions as other VTAM *names*. For information on naming conventions, see "Format of Definition Statements" on page 9.

If you do not specify CKEYNAME, the cryptographic key name defaults to the name on the LU definition statement.

Note: Independent LUs will ignore the CKEYNAME keyword and use their default name.

CLRSESSQ

```
   ┌─CLRSESSQ=NO──────┐
►►─┼──────────────────┼──────────────────────────────────────────►◄
   │                  │
   └─CLRSESSQ=─┬─NO──┬─┘
               └─YES─┘
```

statements:	*PU, LU*
dependencies:	*none*

Specifies whether a dependent LU is forced into an inhibited session state when the LU is disabled. When the LU is enabled, it is returned to normal status. However, sessions are not connected.

CLRSESSQ=NO
Specifies that the LU is not forced into inhibited session state.

CLRSESSQ=YES
Specifies that the LU is forced into inhibited session state. Both active and pending sessions are removed from the session queue.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─CONFGDS=name─────┘
```

statements:	*VBUILD*
dependencies:	*none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─CONFGPW=password─┘
```

statements:	*VBUILD*
dependencies:	*CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

CONNTYPE

```
      ┌─CONNTYPE=value_of_conntype_start_option──────┐
►►────┼──────────────────────────────────────────────┼──────────────────────────►◄
      │              ┌─value_of_conntype_start_option─┐
      └─CONNTYPE=────┼────────────────────────────────┼
                     ├─APPN───────────────────────────┤
                     └─LEN────────────────────────────┘
      ┌─CONNTYPE=APPN─┐   (1)
      │               │
      └─CONNTYPE=APPN─┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*NODETYPE; APPN only*

Specifies for a type 2.1 whether the connection is to be established as a LEN connection or attempted as an APPN connection.

If you do not code the NODETYPE start option, the CONNTYPE operand is ignored, and the connection will be a LEN connection.

CONNTYPE=APPN
Specifies that this connection can support parallel TGs, CP-CP sessions, and CP name change support. When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

CONNTYPE=LEN
Specifies that this connection does not support CP-CP sessions and CP name change, and that VTAM does not check for duplicate CPNAMEs.

Since CONNTYPE=LEN indicates that the PU is not APPN-capable, do not code any APPN-only operands on the same PU statement. If an APPN-only operand is coded with CONNTYPE=LEN, VTAM will issue a message warning of a parameter conflict.

Although you need to code CONNTYPE=APPN for a PU to be an APPN resource, this does not necessarily mean the PU will be APPN; a PU's characteristics are determined at activation. However, coding CONNTYPE=LEN means that the PU can only be a LEN resource, even if, at activation, the XID indicates the PU can be an APPN resource.

COSTBYTE

```
      ┌─COSTBYTE=0──────────────┐
►►────┼─────────────────────────┼──────────────────────────────────────────────►◄
      └─COSTBYTE=cost_per_byte──┘
```

statements:	*PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a cost-per-byte-transmitted to be associated with the transmission group. A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
   ┌─COSTTIME=0──────────────────┐
►►─┤                             ├──────────────────────────────────────────►◄
   └─COSTTIME=cost_per_unit_time─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a cost-per-unit-of-time to be associated with the transmission group. A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

CPCP

```
   ┌─CPCP=value_of_cpcp_start_option──────┐
►►─┤                                      ├──────────────────────────────────►◄
   │        ┌─value_of_cpcp_start_option─┐
   └─CPCP=──┤                            ├──┘
            ├─NO───────────────────────────┤
            └─YES──────────────────────────┘
```

statements: *PU*
dependencies: *APPN only*

Specifies whether CP-CP sessions are supported on this connection. If you do not code this operand, the value defaults to the value specified on the CPCP start option.

CPCP=NO
Indicates CP-CP sessions are not supported on this connection.

CPCP=YES
Indicates CP-CP sessions are supported on this connection.

CPNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─CPNAME=control_point_name─┘
```

statements: *PU*
dependencies: *none*

Identifies the resource's owning control point name. To allow definition of casually connected resources, this operand can be fully qualified with a network ID.

CUADDR

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─CUADDR=device_address────┘
```

statements: *PU*
dependencies: *TRLE*
range: *X'000'–X'FFFF'*

Defines a 3-digit or 4-digit hexadecimal number that identifies the channel device address used when activating the physical unit. The value must match a channel device name supplied when the operating system is generated.

The address must not be enclosed in quotation marks or apostrophes. For example, a channel device address of X'080' should be coded CUADDR=080.

Note: You cannot code CUADDR with the TRLE operand. However, if you omit both the CUADDR and TRLE operands, you must code ISTATUS=INACTIVE and include the address in the U operand in the VARY ACT command that activates the physical unit.

DELAY

```
►►─┬─DELAY=.2──────────────┬──────────────────────────────────────────►◄
   │                       │
   └─DELAY=timeout_value───┘
```

statements: *PU*
dependencies: *none*
range: *.2–9.999*

Specifies the maximum amount of time, in seconds and thousandths of seconds, that VTAM should wait before transmitting low-priority data (that is, PIU transmission priority 0 or 1) to the physical unit.

For more efficient operation, code this value as high as possible. For maximum speed across the interface, set this value to 0. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information.

DISCNT

```
►►─┬─DISCNT=(NO)──────────────────┬──────────────────────────────────►◄
   │                              │
   │           ┌─(NO)──,F─────┐   │
   └─DISCNT=───┴─(YES──┬──────┬──)─┴─
                       └─,NF──┘
```

statements: *PU*
dependencies: *none*

Specifies when VTAM should end its SSCP-LU and SSCP-PU sessions and, when DISCNT=YES, whether to indicate final-use status in the DACTPU request unit when it deactivates a physical unit.

DISCNT=(NO)
If you code NO or use the default, VTAM terminates the SSCP-LU and SSCP-PU sessions when certain conditions are met. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of conditions that can cause sessions to terminate.

DISCNT=(YES)
If you code YES, VTAM automatically terminates the SSCP-LU and SSCP-PU sessions as soon as the last LU-LU session is terminated.

DISCNT=(YES,F)

DISCNT=(YES,NF)
Specifies whether VTAM indicates final-use status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES. This operand does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

If you code F or use the default, final-use status is indicated and the physical connection can be ended. If you code NF, not-final-use status is indicated and the physical connection should not be broken.

Each device has its own requirements regarding final-use status. To determine whether to code F or NF for a given device, consult the appropriate installation publication for the device.

DLOGMOD

```
►►──┬──────────────────────────────────────────────┬──────────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name──┘
```

statements:	*PU, LU*
dependencies:	*MODETAB*

Specifies the name of the default logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the default entry name must be in ISTINCLM. For more information on logon mode entries see “Logon Mode Table” on page 698.

DYNADJCP

```
        ┌─DYNADJCP=value_of_dynadjcp_start_option──────┐
►►──┬───┼──────────────────────────────────────────────┼──────────────────────────►◄
    │   │              ┌─value_of_dynadjcp_start_option─┐│
    │   └─DYNADJCP=────┼────────────────────────────────┼┘│
    │                  ├─NO─────────────────────────────┤ │
    │                  └─YES────────────────────────────┘ │
    │   ┌─DYNADJCP=YES─┐  (1)                            │
    └───┼──────────────┼────────────────────────────────┘
        └─DYNADJCP=YES─┘
```

Notes:

1 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *APPN only; ignored if CPNAME coded*

Specifies whether an ADJCP can be created dynamically for an adjacent node.

Note: This operand is ignored if CPNAME is coded on the physical unit.

DYNADJCP=NO
Indicates that the ADJCP cannot be created dynamically for an adjacent node.

DYNADJCP=YES
Indicates that the ADJCP can be created dynamically for an adjacent node.

DYNLU

```
                 ┌─DYNLU=value_of_dynlu_start_option──────┐
►►───────────────┼────────────────────────────────────────┼──────────────────────────────►◄
                 │          ┌─value_of_dynlu_start_option─┐│
                 └─DYNLU=───┼─────────────────────────────┼┘
                            ├─NO──────────────────────────┤
                            └─YES─────────────────────────┘
                 ┌─DYNLU=YES─┐  (1)
                 │           │
                 └─DYNLU=YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *CDRDYN*

Specifies whether dynamic allocation of CDRSC definitions is supported. If you do not code this operand, the default is the value you coded on the DYNLU start option.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, a resource that was dynamically defined as a CDRSC cannot use that PU to connect to a session.

DYNLU=NO
Specifies that dynamic allocation of CDRSC definitions is not supported. If you code DYNLU=NO and plan to connect to other nodes using CP-CP sessions, then predefine each node in the CDRSC major node. If you do not, CP-CP session setup fails.

DYNLU=YES
Specifies that dynamic allocation of CDRSC definitions is supported.

For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

Note: If you have coded a DYNLU value in the adjacent CP major node for the adjacent CP to which this physical unit is linked, that value overrides the

DYNLU value you code on this PU definition statement. For additional DYNLU considerations, see “DYNLU” on page 21 under the adjacent control point major node.

EAS

```
►►─┬─EAS=256──────────────────┬──────────────────────────────────────────►◄
   └─EAS=number_of_sessions───┘
```

statements:	*PU, LU*
dependencies:	*LOCADDR=0*
range:	*0–65535*

Specifies the estimated number of sessions that will be active with this logical unit at any given time. This operand is valid only for an independent logical unit (LOCADDR=0). If you code EAS for other logical units, VTAM ignores it.

If your estimated value is greater than 256, specify the next higher multiple of 256 for better performance.

You can use this operand in conjunction with the BSBUF buffer pool start option to support peak session levels. See “Buffer Pool” on page 566 for information about BSBUF.

ENCR

```
►►─┬─ENCR=NONE────────────────┬──────────────────────────────────────────►◄
   │                          │
   └─ENCR=─┬─NONE──┬──────────┘
           ├─OPT───┤
           ├─REQD──┤
           └─SEL───┘
```

statements:	*PU, LU*
dependencies:	*none*

Specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

ENCR=NONE
Specifies that this logical unit cannot engage in cryptographic sessions.

ENCR=OPT
Specifies that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from this logical unit. If ENCR=REQD, no sessions can be established with this logical unit unless the host with which it is associated can handle cryptographic sessions.

ENCR=SEL
For local SNA major nodes, SEL has no meaning for logical units. If you specify ENCR=SEL, ENCR=OPT is used instead.

ENCRTYPE

```
►►─┬─ENCRTYPE=DES──────┬──────────────────────────────────────────►◄
   └─ENCRTYPE=TDES24────┘
```

statements: *PU, LU*
dependencies: *none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

Note: The ENCRTYPE value coded on the PU will be sifted down to the appropriate resource (LU).

The SYSDEFed ENCRTYPE of a LU can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to “MODEENT Macroinstruction” on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements: *PU, LU*
dependencies: *TSO sessions only*

Determines how sessions are established for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The PLU acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU's application program, only a single BIND is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) which establish sessions with TSO.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

HPR

```
►►─┬─HPR=YES─────────┬──────────────────────────────────────────────►◄
   │                 │
   └─HPR=──┬─NO──┬───┘
           └─YES─┘
```

statements: *PU*
dependencies: *APPN only*

Specifies whether high performance routing (HPR) support is enabled for a PU.

HPR=YES
Indicates that this PU provides HPR support.

HPR=NO
Indicates that this PU does not provide HPR support.

INCLUD0E

```
►►─┬─INCLUD0E=NO─────────┬──────────────────────────────────────────►◄
   │                     │
   └─INCLUD0E=──┬─NO──┬───┘
                └─YES─┘
```

statements: *PU*
dependencies: *none*

Specifies whether VTAM provides LU names to the logical units defined under this PU. The names are provided by passing a CV0E, Network Name control vector on the ACTLU requests.

INCLUD0E=NO
Indicates that VTAM will not provide LU names to the logical units defined inder this PU.

INCLUD0E=YES
Indicates that VTAM will provide LU names to the logical units defined under this PU.

ISTATUS

```
   ┌─ISTATUS=ACTIVE──────────┐
►►─┼─────────────────────────┼──────────────────────────────────────────────►◄
   │                ┌─ACTIVE─┐│
   └─ISTATUS=───────┼────────┼┘
                    └─INACTIVE─┘
```

statements: *PU, LU*
dependencies: *none*

Determines whether a physical unit or logical unit is to be activated after the first activation of the channel-attachment major node.

ISTATUS=ACTIVE
Specifies that the physical unit or logical unit becomes active when its channel becomes active.

ISTATUS=INACTIVE
Specifies that the physical unit or logical unit remains inactive until the operator activates it.

The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ACT command.

Note: You can dynamically change ISTATUS only on the LU definition statement.

LIMRES

```
   ┌─LIMRES=NO───────────────┐
►►─┼─────────────────────────┼──────────────────────────────────────────────►◄
   │                ┌─NO─┐   │
   └─LIMRES=────────┼────┼───┘
                    └─YES─┘
   ┌─LIMRES=NO─┐   (1)
   │           │
   └─LIMRES=NO─┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *LIMQSINT; LU 6.2 only; PUTYPE=1 or 2*

Specifies whether this physical unit is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes

any sessions that traverse the resource to be deactivated if no conversation is active. A session can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

If you do not specify a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

You can also use LIMRES in conjunction with the LIMINTCP start option to specify the interval to retain a free session using a limited resource for some network management flows (excluding flows using the APPN control point sessions or dependent LU server sessions). If you do not code a value for the LIMINTCP start option, resources are not considered to be limited resources, and the sessions will remain active even after all conversations have ended.

LIMRES=NO
Specifies that this physical unit is not to be treated as a limited resource.

LIMRES=YES
Specifies that this physical unit is to be treated as a limited resource.

LOCADDR

```
►►──LOCADDR=address──────────────────────────────────────────────────►◄
```

statements:	LU
dependencies:	see notes
range:	0–63 (PU type 1) 0–255 (PU type 2)

Specifies the logical unit's local address at the physical unit.

For independent logical units, specify 0. For dependent logical units, specify a non-zero value. All dependent logical units under a given physical unit must have unique LOCADDR values.

An LU definition statement is not required for every possible local address, and LOCADDR values need not be consecutive. Unused local addresses smaller than the largest local address at a station are not assigned network resources.

Notes:

1. The value of LOCADDR depends upon the requirements of the device being defined. Consult the appropriate component description manual for these restrictions.
2. Although you can use this operand to define independent LUs, it is recommended that you either use dynamic definition or predefine the independent LUs as CDRSCs.

LOGAPPL

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu─┘
```

statements: *PU, LU*
dependencies: *NetView or TSO*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before it is used to establish or redrive the session with the controlling primary logical unit.

For independent logical units, the LOGAPPL operand establishes a session with the primary logical unit designated on the LOGAPPL operand if a session does not currently exist with this logical unit. The session is established regardless of whether sessions with other logical units exist. The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────┬──────────────────────────────────────►◄
    └─LOGTAB=interpret_table_name─┘
```

statements: *PU, LU*
dependencies: *dependent LUs only*

Specifies the name of an interpret table that VTAM uses when processing logon requests originating from the logical unit. VTAM uses the interpret table to interpret the name received from an initiation request. See “Interpret Table” on page 690 for a description of how to define an interpret table.

LOGTAB is valid only for dependent logical units.

LUDR

```
    ┌─LUDR=YES──────────┐
►►──┼──────────────────────┼──────────────────────────────────────────►◄
    │          ┌─YES─┐    │
    └─LUDR=────┼─────┼────┘
               └─NO──┘
```

statements: *LU*

dependencies: *none*

Specifies whether an LU is eligible for dynamic reconfiguration.

This operand is ignored for independent LUs.

LUDR=NO
Indicates that VTAM rejects any attempt to move or delete the LU.

LUDR=NO is not valid if PUDR=YES. However, by using DRDS definitions you can create an LU with LUDR=NO under a PU with PUDR=YES, and DR operations on the LU will fail, but DR operations on the PU ignore the LUDR=NO value.

LUDR=YES
Indicates that the LU can be dynamically deleted or moved. Valid only if dynamic reconfiguration is supported for this LU.

LUGROUP

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─LUGROUP=lu_group_name────┘
```

statements: *PU*
dependencies: *none*

Specifies the name of the model LU group that VTAM uses to select a model LU definition when dynamically defining an LU. This operand also indicates that the device supports dynamic definition of LUs.

LUSEED

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─LUSEED=lu_pattern_name───┘
```

statements: *PU*
dependencies: *LUGROUP*

Provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- The first character must be either alphabetical or one of the national characters (@ or $).
- The remaining characters can be either alphabetical, numerical, or national.
- There must be either two or three contiguous national # characters in the name. These characters represent holders, in which the logical unit's local address is substituted when an LU's name is generated. If you specify three # characters, the LU's logical address is substituted in decimal for the # characters. If you specify two # characters, the LU's logical address is substituted in hexadecimal for the # characters.

When used with a user-written SDDLU exit routine, this operand is optional and there are no restrictions on its format. For more information on the SDDLU exit, refer to *OS/390 SecureWay Communications Server: SNA Customization*.

This operand is valid only when you code the LUGROUP operand.

MAXBFRU

```
                (2)
   ┌─MAXBFRU=5──────────────┐
                (1)
   ├─MAXBFRU=1──────────────┤
   ►►──────────────────────────────────────────────────────────►◄
   └─MAXBFRU=number_of_buffers─┘
```

Notes:

1 MAXBFRU defaults to 1 when HPR=NO, XID=NO, CONNTYPE=LEN, or the VTAM node is not HPR capable.

2 MAXBFRU defaults to 5 when HPR=YES, XID=YES, CONNTYPE=APPN, and the VTAM node is HPR capable.

statements:	*PU*
dependencies:	*TRLE*
range:	*1–255*

Specifies the number of buffer units that VTAM uses to receive data from the physical unit. These buffer units are elements of the IOBUF buffer pool.

number_of_buffers is a decimal integer. Always code MAXBFRU such that the *bufsize* value for the IOBUF buffer pool times MAXBFRU is equal to or greater than the maximum RU size used by the controller.

For more information on determining a value for MAXBFRU, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Note: When the TRLE operand is coded on the PU definition statement, the MAXBFRU value coded on the PU definition statement (if coded) is ignored. VTAM uses the MAXBFRU value from the TRLE definition statement in the TRL major node.

MAXDATA

For PU type 1:

```
   ┌─MAXDATA=261──────────────┐
   ►►──────────────────────────────────────────►◄
   └─MAXDATA=number_of_bytes─┘
```

statements:	PU
dependency:	none
range:	5-65535

For PU type 2:

```
   ┌─MAXDATA=265──────────────┐
   ►►──────────────────────────────────────────►◄
   └─MAXDATA=number_of_bytes─┘
```

statements:	PU
dependency:	none
range:	5-65535

Specifies the maximum number of bytes that the physical unit can receive in one PIU or PIU segment, including the transmission header (TH) and request or

response header (RH). To determine the maximum PIU (or PIU segment) size that the physical unit can receive, consult the component publications for the specific type of SDLC station represented by this PU definition statement.

The maximum amount of user data that VTAM sends to the physical unit in one PIU is:
- MAXDATA minus 2 bytes (for a PU type 1)
- MAXDATA minus 6 bytes (for a PU type 2)

Note: The RU part of the PIU contains the user data. When VTAM segments the PIU, the middle and last PIUs contain three more bytes of user data in the RU because there is no RH field. The value for the PIU varies with the device; for example, the 3276 requires MAXDATA=262.

MAXDATA is ignored for type 2.1 nodes attached through an NCP with peripheral node type 2.1 support. The type 2.1 node supplies the actual value used when the connection is established.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information about physical units connected through an NCP.

Note: If the TRLE operand is coded, MAXDATA is ignored. See “TRLE” on page 289 for more information.

Notes:
1. In a local area network, the value for the MAXDATA operand on the PU definition statement must be less than or equal to the value for the MAXDATA operand on the PORT definition statement for the LAN major node to which the switched physical unit is connected.
2. NCP V4R3 changed the MAXDATA calculation for a type 1 physical unit. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on calculating a MAXDATA value for physical units on nonswitched links.

MDLENT

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    └MDLENT=model_name_table_entry_name┘
```

statements: *PU, LU*
dependencies: *MDLTAB*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the associated LU table by default.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────────►◄
    └MDLTAB=model_name_table_name┘
```

statements: *PU, LU*

dependencies: *none*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

The model name is only passed when VTAM initiates the session with the application program, for example, only when the LOGAPPL = parameter is also coded.

If you omit the LOGAPPL operand or the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information on the MODIFY TABLE command.

MODETAB

```
►►─┬─MODETAB=ISTINCLM──────────────┬─────────────────────────────────────────►◄
   └─MODETAB=logon_mode_table_name─┘
```

statements: *PU, LU*
dependencies: *none*

Specifies the name of a logon mode table to be used for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►─┬──────┬──────────────────────────►◄   ►►─name──────────────────────────►◄
   └─name─┘
```

statements: VBUILD *statements:* PU, LU

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node’s status.

On PU: Provides the required minor node name of the physical unit represented by this statement.

On LU: Provides the required minor node name of the logical unit represented by this statement.

NATIVE

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └NATIVE=──┬──NO──┬─┘
              └──YES─┘
```

statements:	*PU*
dependencies:	*BN=YES; APPN only*

Specifies whether this link station represents a connection to a native node.

If NATIVE is not specified on the ADJCP statement or on the PU definition statement, the two nodes negotiate their subnet affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

No subnet negotiations will occur if NATIVE has been specified; in this case the value between the two nodes must match for connection to occur.

If you code NATIVE on both the ADJCP and PU definition statements, the values must match.

For transmission groups (TGs) between nodes, the value specified or negotiated for NATIVE must match. If a TG is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the TG attempting activation is not allowed to become active. If the activation is for a virtual-route-based transmission group then the SSCP-SSCP session will also fail.

NETID

```
►►──┬──────────────────────┬──────────────────────────────────────►◄
    └NETID=network_id──────┘
```

statements:	*PU*
dependencies:	*XNETALS start option*

Specifies a 1–8 character network identifier. This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See “XNETALS” on page 657 for information about XNETALS.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture

specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

NETID=*network_id*
If you specify NETID on the PU definition statement, VTAM ensures that when the physical unit is active, the connecting resource is within the network specified by NETID. The network ID specified by NETID is the CP NETID of the NETID where the PU resides. If you omit NETID, VTAM dynamically acknowledges the network ID during connection establishment and this network ID becomes the CP NETID. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM's network ID.

NN

```
►►──┬──────────────────────────────────────────────────────────────►◄
    └─NN=──┬─NO──┬──┘
           └─YES─┘
```

statements: *PU*
dependencies: *APPN only*

Specifies whether the adjacent node is expected to be a network node.

If you specify a value for NN, it is validated when a connection is attempted to an adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If NN is not specified, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

PACING

```
    ┌─PACING=1──────────────┐
►►──┼───────────────────────┼──────────────────────────────────────►◄
    │           ┌─1─────────┐│
    └─PACING=───┼───────────┼┘
                └─pacing_number─┘
```

statements: *PU, LU*
dependencies: *none*
range: *0–63*

Specifies how VTAM paces the flow of data from the boundary node, which performs pacing for a channel-attached SNA device, to the SLU. When the SLU and the PLU are in the same domain, the PACING value is ignored.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

PDELAY

```
   ┌─PDELAY=TERRESTR──────┐
   │                (1)   │
   ├─PDELAY=NEGLIGIB──────┤
►►─┤                      ├──────────────────────────────────────────────►◄
   │          ┌─TERRESTR─┐│
   └─PDELAY=──┤          ├┘
              ├─LONG─────┤
              ├─NEGLIGIB─┤
              └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

statements:	*PU*
dependencies:	*APPN only*

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PUTYPE

```
►►─┬──────────┬──────────────────────────────────────────────────────────►◄
   └─PUTYPE=2─┘
```

statements:	*PU*
dependencies:	*none*

Specifies the physical unit type. PUTYPE=2 is the only valid value.

REGISTER

For Independent LUs:

```
►►─┬─REGISTER=NO──────────────┬──────────────────────────────►◄
   │                          │
   └─REGISTER=─┬─NO──────────┬┘
               ├─CDSERVR─────┤
               └─NETSRVR─────┘
```

statements: PU, LU
dependency: APPN only

For Dependent LUs:

```
►►─┬─REGISTER=NETSRVR─────────┬──────────────────────────────►◄
   │                          │
   └─REGISTER=─┬─NETSRVR─────┬┘
               ├─CDSERVR─────┤
               └─NO──────────┘
```

statements: PU, LU
dependency: APPN only

Specifies how a resource should be registered. The default depends on whether you code REGISTER for dependent or independent LUs.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

REGISTER=NO
Indicates that the resource should not be registered.

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

RESETLU

```
►►─┬─RESETLU=YES──────────────┬──────────────────────────────►◄
   │                          │
   └─RESETLU=─┬─YES──────────┬┘
              └─NO───────────┘
```

statements: *PU,LU*
dependencies: *dependent LUs only*

Specifies whether a DACTLU request followed by a ACTLU request will be sent during clean-up processing.

RESETLU=YES
Specifies that during clean-up processing for dependent LUs, VTAM will issue an ACTLU request after the DACTLU request.

RESETLU=NO
Specifies that during clean-up processing for LU-LU sessions, RESETLU=NO prevents a DACTLU-ACTLU from flowing to an LU.

SECNET

```
  ┌─SECNET=NO──────┐
►►─┼────────────────┼──────────────────────────────────────────────────►◄
  │                │
  │          ┌─NO──┐
  └─SECNET=──┼─────┼──┘
             └─YES─┘
```

statements: *PU*
dependencies: *valid only for 3174 token-ring attachments*

Specifies whether this physical unit is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

When SECNET=YES, data that is received by VTAM from this physical unit is flagged as requiring special problem-determination considerations when routed to a communication network management (CNM) application.

SECURITY

```
  ┌─SECURITY=UNSECURE──────┐
►►─┼────────────────────────┼──────────────────────────────────────────►◄
  │                        │
  │            ┌─UNSECURE──┐
  └─SECURITY=──┼───────────┼──┘
               ├─PUBLIC────┤
               ├─UNDERGRO──┤
               ├─SECURE────┤
               ├─GUARDED───┤
               ├─ENCRYPT───┤
               └─SHIELDED──┘
```

statements: *PU*
dependencies: *APPN only*

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SPAN

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─SPAN=span_name───────────────┘
```

statements:	*PU, LU*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SSCPFM

```
    ┌─SSCPFM=FSS──────────┐
►►──┼─────────────────────┼──────────────────────────────────────────────►◄
    │          ┌─FSS──┐   │
    └─SSCPFM=──┼──────┼───┘
               └─USSSCS─┘
```

statements:	*PU, LU*
dependencies:	*none*

Determines what type of RUs the logical unit can support in its communications with the SSCP.

SSCPFM=FSS
Specifies that the logical unit supports formatted commands in its communication with the SSCP.

Note: You might receive character-coded messages when SSCPFM=FSS.

SSCPFM=USSSCS
Specifies that the logical unit supports character-coded messages in its communication with the SSCP.

Refer to the publications for each individual device to determine whether formatted or character-coded messages are supported by that device.

TGN

```
    ┌─TGN=ANY─────────────┐
►►──┼─────────────────────┼──────────────────────────────────────────────►◄
    │          ┌─ANY──┐   │
    └─TGN=─────┼──────┼───┘
               └─tg_number─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–20*

Specifies the transmission group (TG) number that represents the connection between this node and the destination APPN node.

By specifying a *tg_number*, you indicate a preferred TG that might be used depending on how the partner nodes negotiate the number.

Coding TGN=ANY, or not coding TGN, allows the transmission group number to be negotiated.

Coding TGN=0 means that the PU does not support parallel TGs. For connections to LEN nodes, or to any node that does not support parallel TGs, TGN=0 is always assumed.

For APPN-capable nodes, the CPNAME and NETID operands are required with the TGN operand. Note that you can specify the CPNAME operand without the TGN operand and the transmission group number will be negotiated. However, if a value for TGN is specified, then a value for CPNAME and NETID must also be specified.

TGP

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─TGP=tgp_name─────┘
```

statements: *PU*
dependencies: *APPN only*

Specifies the name of a transmission group (TG) profile. The characteristics of the TG profile (along with any modifiers specified on this statement) become the characteristics of the PU. If a *tgp_name* is not specified or has not been activated when the PU becomes active, the default transmission group operand values or the override values specified on GROUP, LINE, or PU statements are used.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TOPO

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─TOPO=CONNECT─────┘
```

statements: *PU*
dependencies: *APPN only*

Controls the way APPN connections using this link station will be reported to APPN topology and routing services for inclusion into APPN functions.

Note: Except in the switched major node, the only value for TOPO is CONNECT. If you specify APPN on the CONNTYPE start option, it is not necessary to code TOPO, as you automatically get a value of CONNECT.

TOPO=CONNECT
APPN topology and routing services receives information about the connection and its characteristics when the connection is activated. When the connection is deactivated, APPN topology and routing services receives notification that the TG is inactive.

TRLE

```
►►──┬──────────────────────────┬──────────────────────────────────────────────►◄
    └─TRLE=──┬─trle_name─────┬──┘
             └─*─────────────┘
```

statements: *PU*
dependencies: *CUADDR; APPN only*

Specifies the 8-character name of the TRLE definition statement to be used for the physical unit being defined by this PU definition statement. The TRLE determines the transport characteristics of the PU.

If you specify a value on the TRLE operand that does not have a corresponding TRLE definition statement in the TRL major node, activation of the PU fails.

Note: You cannot code TRLE with the CUADDR operand. However, if you omit both TRLE and CUADDR, you must code ISTATUS=INACTIVE and include the address in the U operand in the VARY ACT command that activates the physical unit.

TYPE

```
►►──TYPE=LOCAL──────────────────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Specifies a local SNA major node. This operand is required.

UPARM1

```
    ┌─UPARM1=128──────────────────────┐
►►──┤                                 ├──────────────────────────────────────►◄
    └─UPARM1=user-defined_operand_value─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
┌─UPARM2=128──────────────────────┐
►►──┤                               ├──────────────────────────────────────►◄
└─UPARM2=user-defined_operand_value─┘
```

statements:	*PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a user-defined operand value.

UPARM3

```
┌─UPARM3=128──────────────────────┐
►►──┤                               ├──────────────────────────────────────►◄
└─UPARM3=user-defined_operand_value─┘
```

statements:	*PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a user-defined operand value.

USSTAB

```
┌─USSTAB=ISTINCDT──────┐
►►──┤                      ├──────────────────────────────────────────────►◄
└─USSTAB=uss_table_name─┘
```

statements:	*PU, LU*
dependencies:	*dependent LUs*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

A terminal user can issue a USS command with the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

USSTAB is valid for dependent logical units only.

VERALSID

Local SNA Major Node

```
      ┌─VERALSID=NO──────┐
►►────┼──────────────────┼──────────────────────────────────────────►◄
      │                  │
      │          ┌─NO─┐  │
      └─VERALSID=┼────┼──┘
                 └─YES┘
      ┌─VERALSID=NO─┐  (1)
      │             │
      └─VERALSID=NO─┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*APPN only*

Specifies whether the adjacent link station name should be used to determine the link station. When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to XID3s. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

VPACING

```
      ┌─VPACING=1──────────────────┐
►►────┼────────────────────────────┼──────────────────────────────────►◄
      └─VPACING=number_of_requests─┘
```

statements:	*PU, LU*
dependencies:	*none*
range:	*0–63*

Specifies how VTAM paces the flow of data from the boundary node containing the PLU to the boundary node that performs pacing for a channel-attached SNA device.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

XID

```
      ┌─XID=NO───────┐
►►────┼──────────────┼──────────────────────────────────────────────►◄
      │        ┌─NO─┐│
      └─XID=───┼────┼┘
               └─YES┘
```

statements:	*PU*
dependencies:	*none*

Specifies the type of channel contact procedure to be used.

XID=NO
Specifies that the local SNA PU type 2 channel contact procedure is to be used.

Note: If TRLE is coded, activation of the PU fails.

XID=YES
Specifies that a type 2.1 channel contact procedure is to be used.

XNETALS

```
      ┌─XNETALS=value_of_XNETALS_start_option─┐
►►────┤                                       ├──────────────────────────────────►◄
      └─XNETALS=──┬─NO──┬──────────────────────┘
                  └─YES─┘
```

statements:	*PU*
dependencies:	*none*

Determines whether the network ID of an attaching node or VTAM's network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true. If the NETID that the PU sent in its XID is not the same as VTAM's and XNETALS=NO or is defaulted, IST680I will be issued.

XNETALS=NO
Indicates that VTAM's network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement, then it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement, then VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES
Indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Note: Providing that XNETALS=YES is coded, you can prohibit nonnative network attachment for LU-LU sessions for a particular PU by coding XNETALS=NO on the GROUP, LINE or PU statements.

LU Group Major Node

To define an LU group major node, code a VBUILD definition statement for the major node, and an LUGROUP definition statement for each model LU group in the major node.

The LU group major node defines one or more model LU groups. Each model LU group contains a list of model LU definitions. VTAM uses the model LU definitions to dynamically define LUs attached to devices that support dynamic definition of dependent LUs.

Each model LU group consists of an LUGROUP definition statement followed by at least one, but no more than 255, LU definition statements. Each LU definition statement represents the LU definitions that are used to dynamically define LUs of a particular device type.

Table 28. LU Group Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LUGROUP	252	R			
name	LUGROUP						
name	LU	ASLENT	243				ASLTAB
		ASLTAB	243				
		CERTIFY	244				
		CKEYNAME	244				
		DLOGMOD	245				MODETAB
		ENCR	245			NONE	
		ENCRTYPE	246				DES
		FASTPASS	246			YES	TSO only
		LOGAPPL	247				NetView or TSO
		LOGTAB	248				dependent LUs
		LUDR	248			YES	

Table 28. LU Group Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		MDLENT	248				MDLTAB
		MDLTAB	249				
		MODETAB	249			ISTINCLM	
		PACING	250			1	
		REGISTER	251			NETSRVR	APPN only
		RESETLU	251		P	YES	
		SSCPFM	252			FSS	
		USSTAB	252			ISTINCDT	dependent LUs
		VPACING	253			2	

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each LU group major node.

LUGROUP Definition Statement

Code an LUGROUP definition statement to specify the start of each model LU group in the LU group major node. All the LU definition statements that follow this definition represent model LUs in the model LU group. The model LU group ends when VTAM encounters either a VBUILD or an LUGROUP definition statement.

LU Definition Statement

Code an LU definition statement for each model LU in the LU group. On type 2.0 and type 2.1 devices that support dynamic definition of dependent LUs, the model LU definitions are used to dynamically define dependent LUs when the device powers on. The maximum number of model LUs that you can define for each LU group is 255.

You cannot take advantage of the sift-down effect in this major node.

Full Syntax

The full syntax for the LU group major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=LUGROUP──────────────────────────────────►◄
    └─name─────┘
```

LUGROUP

```
►►──name──LUGROUP──────────────────────────────────────────────────────────►◄
```

LU

```
►►──name──LU──┬──────────────────────────────────────────────┬──────────────►
              └─ASLENT=associated_lu_table_entry_name─────────┘

►──┬──────────────────────────────────────┬──┬─CERTIFY=NO──────────┬──────►
   └─ASLTAB=associated_lu_table_name──────┘  │                     │
                                             └─CERTIFY=──┬─NO──┬─────┘
                                                         └─YES─┘

►──┬─CKEYNAME=name──────────────┬──────────────────────────────────────────►
   └─CKEYNAME=crypto_key_name───┘

►──┬──────────────────────────────────────────────────────────────────────┬──
   └─DLOGMOD=default_logon_mode_table_entry_name──┬─ENCR=NONE──────────┬──►
                                                  │                     │
                                                  └─ENCR=──┬─NONE──┬─────┘
                                                           ├─OPT───┤
                                                           ├─REQD──┤
                                                           └─SEL───┘

►──┬─ENCRTYPE=DES────┬──┬─FASTPASS=YES────────┬──────────────────────────►
   └─ENCRTYPE=TDES24──┘  │                     │
                         └─FASTPASS=──┬─YES─┬──┘
                                      └─NO──┘

►──┬──────────────────────────────────────┬──┬──────────────────────────────►
   └─LOGAPPL=controlling_primary_lu───────┘  └─LOGTAB=interpret_table_name────┘

►──┬─LUDR=YES────────────┬──┬──────────────────────────────────────────────►
   │                     │  └─MDLENT=model_name_table_entry_name────────────┘
   └─LUDR=──┬─YES─┬──────┘
            └─NO──┘

►──┬──────────────────────────────────────┬──┬─MODETAB=ISTINCLM────────────┬►
   └─MDLTAB=model_name_table_name─────────┘  │                             │
                                              └─MODETAB=logon_mode_table_name┘
```

```
►─┬─PACING=1──────────────┬─┬─REGISTER=NETSRVR──────┬──────────────►
  │                       │ │                       │
  └─PACING=─┬─1──────────┬─┘ └─REGISTER=─┬─NETSRVR─┬─┘
            └─pacing_number─┘             ├─CDSERVR─┤
                                          └─NO──────┘

►─┬─RESETLU=YES──────┬─┬─SSCPFM=FSS──────────┬─┬─USSTAB=ISTINCDT──────────┬──────►
  │                  │ │                     │ │                          │
  └─RESETLU=─┬─YES─┬─┘ └─SSCPFM=─┬─FSS───────┬─┘ └─USSTAB=uss_table_name──┘
             └─NO──┘             ├─SSSCS─────┤
                                 ├─USS3270───┤
                                 └─USS3275───┘

►─┬─VPACING=1──────────────┬──────────────────────────────────────────────►◄
  │                        │
  └─VPACING=number_of_requests─┘
```

Operand Descriptions

The following section describes the operands you can code for the LU group major node.

ASLENT

```
►►──┬─ASLENT=associated_lu_table_entry_name─┬──────────────────────────►◄
```

statements: *LU*
dependencies: *ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the table named in ASLTAB by default.

ASLTAB

```
►►──┬─ASLTAB=associated_lu_table_name─┬──────────────────────────────►◄
```

statements: *LU*
dependencies: *none*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

CERTIFY

```
   ┌─CERTIFY=NO──────────┐
►►─┤                     ├──────────────────────────────────────────►◄
   │              ┌─NO─┐ │
   └─CERTIFY=─────┤    ├─┘
                  └─YES┘
```

statements: *LU*
dependencies: *CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner’s LU support also supports the CERTIFY function.*

Specifies whether VTAM on behalf of the application program will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CKEYNAME

```
   ┌─CKEYNAME=name──────────────┐
►►─┤                            ├──────────────────────────────────────►◄
   └─CKEYNAME=crypto_key_name───┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of the key-encrypting key (KEK) in the cryptographic key data set (CKDS) for the LU. Use CKEYNAME to encipher the session key that is sent in the BIND to the LU when a cryptographic session is established. CKEYNAME has the same format and restrictions as other VTAM *names*. For information on naming conventions, see “Format of Definition Statements” on page 9.

If you do not specify CKEYNAME, the cryptographic key name defaults to the name on the LU definition statement.

Note: Independent LUs will ignore the CKEYNAME keyword and use their default name.

DLOGMOD

```
►►──┬──────────────────────────────────────────────┬──────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name─┘
```

statements:	*LU*
dependencies:	*MODETAB*

Specifies the name of the default logon mode table entry used by default if one is not otherwise provided. If you do not code this operand, and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on logon mode entries, see “Logon Mode Table” on page 698.

ENCR

```
      ┌─ENCR=NONE──────────┐
►►────┼────────────────────┼──────────────────────────────────────────────────►◄
      │            ┌─NONE─┐│
      └─ENCR=──────┼──────┼┘
                   ├─OPT──┤
                   ├─REQD─┤
                   └─SEL──┘
```

statements:	*LU*
dependencies:	*none*

Tells whether this logical unit has any special requirements for enciphering and deciphering messages.

ENCR=NONE
Specifies that this logical unit cannot engage in cryptographic sessions.

ENCR=OPT
Specifies that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from this logical unit. If ENCR=REQD, no sessions can be established with this logical unit unless the host with which it is associated is able to handle cryptographic sessions.

ENCR=SEL
For an LU group major node, SEL has no meaning for logical units. If you specify ENCR=SEL, ENCR=OPT is used instead.

ENCRTYPE

```
►►─┬─ENCRTYPE=DES──────┬──────────────────────────────────────────►◄
   └─ENCRTYPE=TDES24────┘
```

statements: *LU*
dependencies: *none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

The SYSDEFed ENCRTYPE of a LU can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to “MODEENT Macroinstruction” on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements: *LU*
dependencies: *TSO sessions only*

Determines how session establishment is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The PLU acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU's application program, only a single BIND is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) that establish sessions with TSO.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

LOGAPPL

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu─┘
```

statements:	*LU*
dependencies:	*NetView or TSO*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

For independent logical units, the LOGAPPL operand establishes a session with the primary logical unit designated on the LOGAPPL operand if a session does not currently exist with this logical unit. The session is established regardless of whether sessions with other logical units exist. The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Notes:

1. If an error occurs for the switched connection (INOP), VTAM reallocates the LOGAPPL session and attempts to redial the logical unit to recover the connection. In other words, the session will be terminated by breaking the dial connection instead of by normal session termination using LOGOFF and TERMSELF.
2. If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─LOGTAB=interpret_table_name──┘
```

statements: *LU*
dependencies: *valid only for dependent LUs*

Specifies the name of an interpret table that VTAM uses when processing logon requests originating from the logical unit. VTAM uses the interpret table to interpret the name received from an initiation request. See “Interpret Table” on page 690 for a description of how to define an interpret table.

LOGTAB is valid for dependent logical units only.

LUDR

```
      ┌─LUDR=YES──────┐
►►────┼───────────────┼──────────────────────────────────────────────►◄
      │       ┌─YES─┐ │
      └─LUDR=─┼─────┼─┘
              └─NO──┘
```

statements: *LU*
dependencies: *none*

Specifies whether an LU is eligible for dynamic reconfiguration.

This operand is ignored for independent LUs.

LUDR=NO
Specifies that an LU cannot be dynamically deleted or moved.

LUDR=YES
Specifies that an LU can be dynamically deleted or moved.

MDLENT

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name──┘
```

statements: *LU*
dependencies: *MDLTAB*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─┘
```

statements: *LU*
dependencies: *none*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

If you omit the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MODETAB

```
      ┌─MODETAB=ISTINCLM──────────────┐
►►────┤                               ├──────────────────────────────────────►◄
      └─MODETAB=logon_mode_table_name─┘
```

statements: *LU*
dependencies: *none*

Specifies the name of a logon mode table to be used for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►──┬──────┬──────────────────────►◄   ►►──name──────────────────────────►◄
    └─name─┘
```

statements: VBUILD *statements:* LUGROUP, LU

Specifies the name of the resource being defined.

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On LUGROUP: Provides the required name of the model LU group.

On LU: Provides the required name that VTAM uses when selecting the model LU group that it will use to build dynamic LU definitions. *name* is matched against the machine type and model number (model acronym). The control unit provides VTAM with the model acronym when the device powers on.

Each *name* within a model LU group must be unique. However, different model LU groups can contain model LU definitions with the same name.

For more information on model acronyms and how VTAM matches a model acronym to a model LU definition, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

The following are valid characters:
- Alphabetical (A—Z)
- Numerical (0–9)
- National (@, #, or $)

When matching an LU acronym to an LU definition, the @ character has the following special meanings:
- Within a name, @ represents a single unidentified character.
- If @ is the last character in a name, it represents any number of unknown characters.
- If @ is the only character used in a name, it will match any LU acronym.

PACING

```
►►─┬─PACING=1──────────────────┬──────────────────────────────────────────►◄
   │                           │
   └─PACING=─┬─1──────────────┬┘
             └─pacing_number──┘
```

statements:	*LU*
dependencies:	*none*
range:	*0–63*

Specifies how VTAM paces the flow of data between the logical unit and the boundary node to which it is connected. When the SLU and the PLU are in the same domain, the PACING value is ignored.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

REGISTER

```
►►─┬─REGISTER=NETSRVR──────────┬──────────────────────────────────────►◄
   │                           │
   └─REGISTER=─┬─NETSRVR─┬─────┘
               │         │
               ├─CDSERVR─┤
               └─NO──────┘
```

statements: *LU*
dependencies: *APPN only*

Specifies how a resource should be registered. The default depends on whether you code REGISTER for dependent or independent LUs. Because the LU group major node is used to define only dependent LUs, the default is NETSRVR.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

REGISTER=NO
Indicates that the resource should not be registered.

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

RESETLU

```
►►─┬─RESETLU=YES───────────┬──────────────────────────────────────►◄
   │                       │
   └─RESETLU=─┬─YES─┬──────┘
              │     │
              └─NO──┘
```

statements: *PU, LU*
dependencies: *dependent LUs only*

Specifies whether a DACTLU request followed by a ACTLU request will be sent during clean-up processing.

RESET=YES
Specifies that during clean-up processing for dependent LUs, VTAM will issue an ACTLU request after the DACTLU request.

RESETLUR=NO
Specifies that during clean-up processing for LU-LU sessions, RESETLU=NO prevents a DACTLU-ACTLU from flowing to an LU.

SSCPFM

```
►►─┬─SSCPFM=FSS──────────┬──────────────────────────────────────────►◄
   │                     │
   └─SSCPFM=─┬─FSS────┬──┘
             │        │
             ├─SSSCS──┤
             ├─USS3270┤
             └─USS3275┘
```

statements: *LU*
dependencies: *none*

Determines what type of RUs the logical unit can support in its communications with the SSCP.

SSCPFM=FSS
Specifies that field-formatted messages are supported.

SSCPFM=USSSCS
Specifies that character-coded commands are supported.

SSCPFM=USS3270
Allows printing of a user-defined logon message (USSMSG10) on BSC printers.

SSCPFM=USS3275
Prevents printing of a user-defined logon message (USSMSG10) for a terminal defined as an SDLC 3275 (PU type 1) that might have an attached printer. Code SSCPFM=USS3275 to prevent the user-defined logon message from printing on BSC printers.

TYPE

```
►►─TYPE=LUGROUP──────────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Defines an LU group major node to VTAM. VTAM uses the model LU groups to dynamically define dependent LUs attached to type 2.0 and type 2.1 devices that support dynamic definition of dependent LUs.

USSTAB

```
►►─┬─USSTAB=ISTINCDT──────┬──────────────────────────────────────────►◄
   │                      │
   └─USSTAB=uss_table_name┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

A terminal user can issue a USS command with the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

USSTAB is valid for dependent logical units only.

VPACING

```
►►─┬─VPACING=1──────────────────┬──────────────────────────────────────►◄
   └─VPACING=number_of_requests─┘
```

statements:	*LU*
dependencies:	*none*
range:	*0–63*

Determines how VTAM paces the flow of data from VTAM to the boundary node to which the logical unit is connected.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Model Major Node

To define dynamic switched PU and LU definitions in a model major node, code a VBUILD definition statement, followed by PU and LU definition statements. The PU and LU definitions contained within this node are used to build dynamic switched definitions for previously undefined devices.

Table 29. Model Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=MODEL	291	R			
name	PU	ADDR	261	R			nonswitched connections
		ANS	261			STOP	
		AUTHLEN	263			YES	APPN only; Type 2.1
		CAPACITY	264			8K	APPN only
		CONNTYPE	266			CONNTYPE start option	APPN only
		COSTBYTE	267			0	APPN only
		COSTTIME	267			0	APPN only
		CPCP	267			CPCP start option	APPN only

Table 29. Model Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		DATMODE	267			HALF	
		DISCNT	268			(NO)	
		DYNADJCP	269			DYNADJCP start option	APPN only
		DYNLU	270			DYNLU start option	CDRDYN
		INCLUD0E	274			NO	
		IRETRY	274			NO	
		LIMRES	275			NO	LIMQSINT; LU 6.2; PUTYPE=1, 2
		LUGROUP	277				
		LUSEED	277				LUGROUP
		MAXDATA	278			261 or 265	
		MAXOUT	279			1	
		MAXSESS	279				NCP-attached independent LUs
		NATIVE	281				BN=YES APPN only
		NETID	229				XNETALS
		NN	282				APPN only
		PASSLIM	283			1	
		PDELAY	284			TERRESTR	APPN only
		PUTYPE	284			2	
		SECNET	286			NO	
		SECURITY	287			UNSECURE	APPN only
		SPAN	287				NetView
		TGP	289				APPN only
		TOPO	289			CONNECT	APPN only
		TRLE	289				XCF connectivity only
		UPARM1	291			128	APPN only
		UPARM2	291			128	APPN only
		UPARM3	292			128	APPN only
		VERALSID	292			NO	APPN only
		VTAMTOPO	293				
		XNETALS	294			XNETALS start option value	
name	LU	LOCADDR	275	R			see description
		ASLENT	262				ASLTAB
		ASLTAB	262				
		BATCH	263			NO	Pre-V4R3 NCP only
		CERTIFY	264			NO	ENCR
		CKEYNAME	265				
		CLRSESSQ	265				
		DLOGMOD	269				MODETAB
		ENCR	271			NONE	
		ENCRTYPE	271				DES
		FASTPASS	272			YES	TSO only

Table 29. Model Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		FEATUR2	273			(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	
		LOGAPPL	276				NetView or TSO
		LOGTAB	276				
		MDLENT	279				MDLTAB
		MDLTAB	280				
		MODETAB	280			ISTINCLM	
		PACING	283			1	
		REGISTER	285			see description	
		RESETLU	286			YES	
		SPAN	287				NetView
		SSCPFM	288			FSS	
		TERM	288				
		USSTAB	292			ISTINCDT	dependent LUs
		VPACING	293			2	

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each model major node.

PU Definition Statement

Code a PU definition statement for each physical unit in the model major node. The model PU definition statement is used to create a PU when the installation exit routine returns the name of the model and the name of the previously undefined PU to VTAM.

When defining a model major node, refer to *OS/390 SecureWay Communications Server: SNA Customization* for information on using the NIDDEF file. The exit routine is required.

LU Definition Statement

Code an LU definition statement for each logical unit in the model major node. The model LU definition statement is used to create LUs when the installation exit routine returns the name of the model and the name of the previously undefined LU to VTAM.

Note: You cannot take advantage of the sift-down effect in this major node.

NODETYPE Start Option

Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

Full Syntax

The full syntax for the model major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=MODEL──────────────────────────────────────►◄
    └─name─────┘
```

PU

```
►►──name──PU──ADDR=link_station_address──┬─ANS=STOP──────────────┬──────────►
                                         │                       │
                                         └─ANS=──┬─STOP────────┬─┘
                                                 └─CONTINUE────┘

►──┬─AUTHLEN=YES──────────┬──┬─CAPACITY=8K──────────────┬──────────────────►
   │                      │  │                          │
   └─AUTHLEN=──┬─NO───┬─────┘  └─CAPACITY=number_of_bits─┘
               └─YES──┘

►──┬─CONNTYPE=value_of_conntype_start_option──────────────┬──────────────────►
   │                                                      │
   │                  ┌─value_of_conntype_start_option──┐ │
   │                  │                                 │ │
   └─CONNTYPE=────────┼─APPN────────────────────────────┼─┘
                      └─LEN─────────────────────────────┘
   ┌─CONNTYPE=APPN──┐  (1)
   │                │
   └─CONNTYPE=APPN──┘

►──┬─COSTBYTE=0──────────────┬──┬─COSTTIME=0──────────────────┬──────────►
   │                         │  │                             │
   └─COSTBYTE=cost_per_byte──┘  └─COSTTIME=cost_per_unit_time─┘
```

```
►─┬─CPCP=value_of_cpcp_start_option──────────┬─┬─DATMODE=HALF──────────┬──►
  │                                          │ │                       │
  └─CPCP=─┬─value_of_cpcp_start_option─┬─────┘ └─DATMODE=─┬─HALF─┬────┘
          ├─NO──────────────────────────┤                  └─FULL─┘
          └─YES─────────────────────────┘

►─┬─DISCNT=(NO)──────────────────────────┬──────────────────────────────►
  │                                      │
  └─DISCNT=─┬─(NO)──,F──────────┬─────────┘
            └─(YES─┬──────────┬─)─────────┘
                   └─,NF──────┘

►─┬─┬─DYNADJCP=value_of_dynadjcp_start_option──────────┬──────────────►
  │ │                                                  │
  │ └─DYNADJCP=─┬─value_of_dynadjcp_start_option─┬────┘
  │             ├─NO──────────────────────────────┤
  │             └─YES─────────────────────────────┘
  │ ┌─DYNADJCP=YES──┐  (2)
  │ │               │
  │ └─DYNADJCP=YES──┘
  └──────────────────────────────────────────────────────────────────┘

►─┬─┬─DYNLU=value_of_dynlu_start_option──────────┬─┬─INCLUDOE=NO──────┬──►
  │ │                                            │ │                  │
  │ └─DYNLU=─┬─value_of_dynlu_start_option─┬─────┘ └─INCLUDOE=─┬─NO─┬─┘
  │          ├─NO──────────────────────────┤                   └─YES┘
  │          └─YES─────────────────────────┘
  │ ┌─DYNLU=YES──┐  (3)
  │ │            │
  │ └─DYNLU=YES──┘
  └──────────────────────────────────────────────────────────────────┘

►─┬─IRETRY=NO──────────┬─┬─┬─LIMRES=NO──────────┬─┬──────────────────►
  │                    │ │ │                    │ │
  └─IRETRY=─┬─NO─┬─────┘ │ └─LIMRES=─┬─NO─┬─────┘ │
            └─YES┘       │           └─YES┘       │
                         │ ┌─LIMRES=NO──┐  (4)    │
                         │ │            │         │
                         │ └─LIMRES=NO──┘         │
                         └────────────────────────┘

►─┬──────────────────────────────────────────────┬──────────────────────►
  │                                              │
  ├─LUGROUP=lu_group_name────────────────────────┤
  └─LUGROUP=lu_group_name,LUSEED=lu_pattern_name─┘

                 (6)
►─┬─MAXDATA=265──────────────┬─┬─MAXOUT=1──────────────────┬──────────►
  │                          │ │                           │
  │                 (5)      │ │                           │
  ├─MAXDATA=261──────────────┤ │                           │
  │                          │ │                           │
  └─MAXDATA=number_of_bytes──┘ └─MAXOUT=max_number_of_pius─┘
```

Model Major Node

```
►──┬──────────────────────────────────┬──┬──────────────────┬──────────────────────────────────►
   └MAXSESS=max_number_LU-LU_sessions─┘  └NATIVE=──┬──NO──┬─┘
                                                   └──YES─┘

                                                   ┌──PASSLIM=1──────────────┐
►──┬──────────────┬──┬──────────┬──┬──────────┬──┼─────────────────────────┼──────────────────►
   └NETID=network_id─┘  └NN=──┬──NO──┬─┘  └──────────────────────────────┘
                              └──YES─┘

   ┌──PDELAY=TERRESTR──────────┐
   │                   (7)     │
   ├──PDELAY=NEGLIGIB──────────┼──┬──PUTYPE=2──────────┬──┬──SECNET=NO──────────┬──────────────►
   │                           │  │                    │  │                     │
   │          ┌──TERRESTR──┐   │  │          ┌──2──┐   │  │          ┌──NO──┐   │
   └──PDELAY=─┼────────────┼───┘  └──PUTYPE=─┼─────┼───┘  └──SECNET=─┼──────┼───┘
              ├──LONG───────┤                └──1──┘                  └──YES─┘
              ├──NEGLIGIB───┤      ┌──PUTYPE=2──┐  (8)
              └──PACKET─────┘      └──PUTYPE=2──┘

   ┌──SECURITY=UNSECURE────────┐
   │                           │  ┌──SPAN=span_name──┐
   │          ┌──UNSECURE──┐   │  │                  │
   └──SECURITY=─┼────────────┼───┘  └──────────────────┘
              ├──PUBLIC─────┤
              ├──UNDERGRO───┤
              ├──SECURE─────┤
              ├──GUARDED────┤
              ├──ENCRYPT────┤
              └──SHIELDED───┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

2 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

3 DYNLU=YES is the only valid value if the TRLE operand is coded.

4 LIMRES=NO is the only valid value if the TRLE operand is coded.

5 MAXDATA defaults to 261 when PUTYPE=1 is coded.

6 MAXDATA defaults to 265 when PUTYPE=2 is coded.

7 NEGLIGIB is the default if the TRLE operand is coded.

8 PUTYPE=2 is the only valid value if the TRLE operand is coded.

```
►►──┬──────────────┬──┬──────────────┬──┬──────────────────────────┬──────────────────────────►
    └TGP=tgp_name──┘  └TOPO=CONNECT──┘  └TRLE=──┬──trle_name──┬──┘
                                                └──*──────────┘

   ┌──UPARM1=128──────────────────────┐  ┌──UPARM2=128──────────────────────┐
►──┼──────────────────────────────────┼──┼──────────────────────────────────┼──────────────►
   └──UPARM1=user-defined_operand_value┘  └──UPARM2=user-defined_operand_value┘
```

```
►─┬─UPARM3=128──────────────────────────┬─┬─VERALSID=NO──────────┬──────────────►
  │                                     │ │                      │
  └─UPARM3=user-defined_operand_value─┘ │ └─VERALSID=─┬─NO──┬─┘
                                        │             └─YES─┘
                                        │ ┌─VERALSID=NO──┐  (1)
                                        │ │              │
                                        │ └─VERALSID=NO─┘
```

```
►──┬──────────────────────────────────┬─┬─XNETALS=value_of_XNETALS_start_option─┬──►◄
   │                                  │ │                                       │
   └─VTAMTOPO=─┬─REPORT───┬─┘ │ └─XNETALS=─┬─NO──┬──────────────────────┘
               │          │   │            └─YES─┘
               └─NOREPORT─┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

LU

```
►►─name──LU──LOCADDR=address──┬──────────────────────────────────────────┬──►
                              └─ASLENT=associated_lu_table_entry_name─┘
```

```
►──┬──────────────────────────────────┬─┬─BATCH=NO──────────┬──────────────►
   │                                  │ │                   │
   └─ASLTAB=associated_lu_table_name─┘ │ └─BATCH=─┬─NO──┬─┘
                                       │          └─YES─┘
```

```
►──┬─CERTIFY=NO──────────┬──┬─CKEYNAME=name──────────────┬──┬─CLRSESSQ=NO──────────┬──►
   │                     │  │                            │  │                      │
   └─CERTIFY=─┬─NO──┬─┘  │  └─CKEYNAME=crypto_key_name─┘  │  └─CLRSESSQ=─┬─NO──┬─┘
              └─YES─┘                                      │              └─YES─┘
```

```
►──┬──────────────────────────────────────────────┬─┬─ENCR=NONE──────────┬──────►◄
   │                                              │ │                    │
   └─DLOGMOD=default_logon_mode_table_entry_name─┘ │ └─ENCR=─┬─NONE──┬─┘
                                                    │         ├─OPT───┤
                                                    │         ├─REQD──┤
                                                    │         └─SEL───┘
```

```
►►─┬─ENCRTYPE=DES────┬──┬─FASTPASS=YES──────────┬──────────────────────────────►
   │                  │  │                       │
   └─ENCRTYPE=TDES24─┘  │  └─FASTPASS=─┬─YES──┬─┘
                        │              └─NO───┘
```

Model Major Node

```
►─┬─FEATUR2=(DUALCSE,MODEL1,NOEDATS,NOPRINTR,NOSELPEN)─┬──────────────────────────►
  └─FEATUR2=(──┬─┬─DUALCSE──┬─┬──────────────────────┬─)──┘
               │ └─LOWERCSE─┘ │                      │
               │ ┌─MODEL1───┐ │                      │
               │ └─MODEL2───┘ │                      │
               │ ┌─NOEDATS──┐ │                      │
               │ └─EDATS────┘ │                      │
               │ ┌─NOPRINTR─┐ │                      │
               │ └─PRINTR───┘ │                      │
               │ ┌─NOSELPEN─┐ │                      │
               └─└─SELPEN───┘─┘

►─┬──────────────────────────────────────┬─┬──────────────────────────────────────┬──►
  └─LOGAPPL=controlling_primary_lu───────┘ └─LOGTAB=interpret_table_name──────────┘

►─┬──────────────────────────────────────────┬─┬──────────────────────────────────────────┬──►
  └─MDLENT=model_name_table_entry_name───────┘ └─MDLTAB=model_name_table_name──────────────┘

►─┬─MODETAB=ISTINCLM──────────────────────┬─┬─PACING=1──────────────────────────┬──►
  └─MODETAB=logon_mode_table_name─────────┘ │                                   │
                                             └─PACING=──┬─1──────────────────────┬─┘
                                                        └─pacing_number──────────┘

                 (2)
  ┌─REGISTER=NO──────────────────────┐
                      (1)
  ├─REGISTER=NETSRVR─────────────────┤   ┌─RESETLU=YES──────────┐
  │                                  │   │                      │
  │              ┌─NO──────────┐     │   │                      │   └─SPAN=span_name──────────┘
  │              ├─NETSRVR─────┤     │   └─RESETLU=──┬─YES──┬───┘
  └─REGISTER=────┴─CDSERVR─────┴─────┘               └─NO───┘
```

Notes:

1 REGISTER defaults to NETSRVR for dependent LUs.

2 REGISTER defaults to NO for independent LUs.

```
►►─┬─SSCPFM=FSS──────────────────────┬─┬──────────────────────────────────────────────►
   │                                 │ │
   └─SSCPFM=──┬─FSS──────────────────┬─┘ └─TERM=──┬─3275──────────────────┬──┬──┘
              ├─USSNTO───────────────┤            ├─3277──────────────────┤  │
              ├─USSSCS───────────────┤            ├─3284──────────────────┤  │
              ├─USS3270──────────────┤            ├─3286──────────────────┤  │
              ├─USS3275──────────────┤            └─terminal_Type─────────┘  │
              └─USS3780──────────────┘
```

```
►─┬─USSTAB=ISTINCDT──────────┬─┬─VPACING=2──────────────────────┬──────────────►◄
  └─USSTAB=uss_table_name────┘ └─VPACING=─┬─2──────────────────┬─┘
                                          └─number_of_requests─┘
```

Operand Descriptions

The following section describes the operands you can code for the model major node.

ADDR

```
►►──ADDR=link_station_address──────────────────────────────────────────────►◄
```

statements:	*PU*
dependencies:	*nonswitched connections*

Specifies the hexadecimal SDLC station address for the physical unit. You can specify one or two hexadecimal digits for ADDR. For example, if the address is X'03', you can code ADDR=03 or ADDR=3.

This address is required for all physical units attached through SDLC lines and X.25 networks.

Notes:

1. This operand is ignored for switched subarea connections.
2. If the TRLE operand is coded, ANS is ignored. See “TRLE” on page 289 for more information.

ANS

```
►►─┬─ANS=STOP─────────────────┬──────────────────────────────────────────────►◄
   └─ANS=─┬─STOP─────────────┬─┘
          └─CONTINUE─────────┘
```

statements:	*PU*
dependencies:	*none*

Determines whether the physical unit continues to operate when the NCP enters automatic network shutdown. The value coded on this definition statement overrides whatever value is coded on the ANS definition statement for the NCP.

For switched subarea connections, this operand specifies whether a switched subarea connection over lines attached to this physical unit should remain active or be dropped when the owning SSCP is lost. This is used only on a PU definition statement that defines a subarea physical unit that will be connected to a link station within an NCP.

Notes:

1. This operand is ignored for switched subarea connections.
2. If the TRLE operand is coded, ANS is ignored. See “TRLE” on page 289 for more information.

ANS=CONTINUE or CONT

Indicates that LU-LU sessions will continue when the NCP enters automatic network shutdown.

You must code ANS=CONTINUE or CONT if the terminal is to participate in XRF sessions.

When a session is continued and takeover occurs, the dependent logical unit taken over must support ACTLU(ERP) or the session is terminated during takeover processing. An independent logical unit does not receive ACTLU, so this restriction does not apply.

Note that there are several restrictions for session continuity in certain configurations. For information on these restrictions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

ANS=STOP

Indicates that LU-LU sessions will be terminated when the NCP enters automatic network shutdown for all LUs associated with this PU.

ASLENT

```
►►──┬──────────────────────────────────────────┬──────────────────────────────────►◄
    └─ASLENT=associated_lu_table_entry_name────┘
```

statements: *LU*
dependencies: *ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►──┬──────────────────────────────────────────┬──────────────────────────────────►◄
    └─ASLTAB=associated_lu_table_name──────────┘
```

statements: *LU*
dependencies: *none*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can

provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

AUTHLEN

```
►►─┬─AUTHLEN=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─AUTHLEN=──┬─NO──┬────┘
               └─YES─┘
```

statements: *PU*
dependencies: *Type 2.1; APPN only*

Specifies whether VTAM should pass the transmission priority field specified by this PU to another PU.

Note: If the TRLE operand is coded, AUTHLEN is ignored. See “TRLE” on page 289 for more information.

AUTHLEN=NO
VTAM will not pass the specified transmission priority, instead the default priority of medium will be passed.

AUTHLEN=YES
VTAM will pass the specified transmission priority.

BATCH

```
►►─┬─BATCH=NO─────────────┬──────────────────────────────────────────►◄
   │                      │
   └─BATCH=──┬─NO──┬──────┘
             └─YES─┘
```

statements: *LU*
dependencies: *pre-V4R3 NCPs only*

Specifies the processing priority that the NCP uses for the logical unit.

BATCH=NO
Specifies a high priority (suitable for interactive application programs).

BATCH=YES
Specifies a low priority.

Note: This operand applies to pre-V4R3 NCPs only. If you code the BATCH operand and the NCP is V4R3 or later, the operand is ignored.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information about this operand.

CAPACITY

```
►►─┬─CAPACITY=8K──────────────┬──────────────────────────────────────►◄
   │                          │
   └─CAPACITY=number_of_bits──┘
```

statements:	*PU*
dependencies:	*APPN only*
range:	*1K–1000M*

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either kilobits per second (for example, 100K) or megabits per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Note: Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CERTIFY

```
►►─┬─CERTIFY=NO──────────┬──────────────────────────────────────►◄
   │                     │
   │              ┌─NO─┐ │
   └─CERTIFY=─────┴────┴─┘
                  └─YES─┘
```

statements:	*LU*
dependencies:	*CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner's LU support also supports the CERTIFY function.*

Specifies whether VTAM on behalf of the application program will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CKEYNAME

```
┌─CKEYNAME=name──────────────┐
►►─┤                          ├──────────────────────────────────────────►◄
└─CKEYNAME=crypto_key_name─┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of the key-encrypting key (KEK) in the cryptographic key data set (CKDS) for the LU. Use CKEYNAME to encipher the session key that is sent in the BIND to the LU when a cryptographic session is established. CKEYNAME has the same format and restrictions as other VTAM *names*. For information on naming conventions, see “Format of Definition Statements” on page 9.

If you do not specify CKEYNAME, the cryptographic key name defaults to the name on the LU definition statement.

Note: Independent LUs will ignore the CKEYNAME keyword and use their default name.

CLRSESSQ

```
┌─CLRSESSQ=NO──────────┐
►►─┤                      ├──────────────────────────────────────────►◄
│          ┌─NO──┐      │
└─CLRSESSQ=─┤     ├──────┘
└─YES─┘
```

statements: *LU*

Specifies whether a dependent LU is forced into an inhibited session state when the LU is disabled. When the LU is enabled, it is returned to normal status. Sessions, however, are not connected.

CLRSESSQ=YES
Specifies that the LU is forced into an inhibited session state. Both active and pending sessions are removed from the session queue.

CLRSESSQ=NO
Specifies that the LU is not forced into inhibited session state. This is the default.

CONNTYPE

```
      ┌─CONNTYPE=value_of_conntype_start_option──────┐
►►────┼──────────────────────────────────────────────┼──────────────────────────────►◄
      │              ┌─value_of_conntype_start_option─┐
      └─CONNTYPE=────┼────────────────────────────────┼──┘
                     ├─APPN───────────────────────────┤
                     └─LEN────────────────────────────┘
      ┌─CONNTYPE=APPN─┐  (1)
      │               │
      └─CONNTYPE=APPN─┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*APPN only*

Specifies for a type 2.1 whether the connection is to be established as a LEN connection or attempted as an APPN connection.

If you do not code the NODETYPE start option, the CONNTYPE operand is ignored, and the connection will be a LEN connection.

Note: If the TRLE operand is coded, CONNTYPE must be specified as APPN or allowed to default. See “TRLE” on page 289 for more information.

CONNTYPE=APPN
Specifies that this connection can support parallel TGs, CP-CP sessions, and CP name change support.

When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

CONNTYPE=LEN
Specifies that this connection does not support CP-CP sessions and CP name change, and that VTAM does not check for duplicate CPNAMEs.

Since CONNTYPE=LEN indicates that the PU is not APPN-capable, do not code any APPN-only operands on the same PU statement. If an APPN-only operand is coded with CONNTYPE=LEN, VTAM will issue a message warning of a parameter conflict.

Although you need to code CONNTYPE=APPN for a PU to be an APPN resource, this does not necessarily mean the PU will be APPN; a PU’s characteristics are determined at activation. However, coding CONNTYPE=LEN means that the PU can only be a LEN resource, even if, at activation, the XID indicates the PU can be an APPN resource.

COSTBYTE

```
   ┌─COSTBYTE=0──────────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   └─COSTBYTE=cost_per_byte──┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a cost-per-byte-transmitted to be associated with the TG. A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
   ┌─COSTTIME=0──────────────────┐
►►─┤                             ├──────────────────────────────────────────►◄
   └─COSTTIME=cost_per_unit_time─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a cost-per-unit-time to be associated with the transmission group (TG). A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

CPCP

```
   ┌─CPCP=value_of_cpcp_start_option──────┐
►►─┤                                      ├──────────────────────────────────►◄
   │          ┌─value_of_cpcp_start_option─┐
   └─CPCP=────┤                            ├
              ├─NO───────────────────────────┤
              └─YES──────────────────────────┘
```

statements: *PU*
dependencies: *APPN only*

Specifies whether CP-CP sessions are supported on this connection. If you do not code CPCP, the value defaults to YES only if you specified YES for the CPCP start option. Otherwise, the value defaults to NO.

DATMODE

```
   ┌─DATMODE=HALF──────┐
►►─┤                   ├──────────────────────────────────────────────────►◄
   │          ┌─HALF─┐ │
   └─DATMODE=─┤      ├─┘
              └─FULL─┘
```

statements: *PU*
dependencies: *none*

Specifies whether the physical unit communicates in half-duplex or full-duplex data mode. The associated line must support simultaneous transmission in both directions.

Note: If the TRLE operand is coded, DATMODE is ignored. See “TRLE” on page 289 for more information.

DATMODE=FULL
Indicates that the physical unit can communicate in full-duplex data mode.

DATMODE=HALF
Indicates that the physical unit can communicate in half-duplex data mode.

DISCNT

```
  ┌─DISCNT=(NO)──────────────┐
►►┼──────────────────────────┼──────────────────────────────────────►◄
  │          ┌─(NO)──,F──────┐│
  └─DISCNT=──┼─(YES──────────┼)┘
             └──,NF──┘
```

statements: *PU*
dependencies: *none*

Specifies when VTAM should end its SSCP-LU and SSCP-PU sessions and, when DISCNT=YES, whether to indicate final-use status in the DACTPU request unit when it deactivates a physical unit.

DISCNT=(NO)

DISCNT=(YES)
Specifies whether VTAM physically disconnects the physical unit when the last logical unit ends its session with its application program (that is, when there are no more LU-LU sessions).

For a physical unit on a switched link, disconnection means that the dial connection is ended (in effect, the telephone is hung up, thus saving telephone charges) and the SSCP-PU session is terminated. Disconnection on a switched link does not involve deactivating the physical unit or its logical units. (That is, sessions can be requested with those logical units, causing the physical connection to be re-established.)

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about disconnection.

Note: When the model major node is used to define a dynamic destination LU on a switched connection, the definition for this LU is placed in the dynamic switched major node, ISTDSWMN. DISCNT affects that LU definition.

If DISCNT=NO, the definitions for the PU and LU will remain in ISTDSWMN, even after all LU sessions have ended, as long as the PU remains active.

IF DISCNT=YES, then the PU and LU definition statements are erased from ISTDSWMN after all sessions have ended.

DISCNT=(YES,F)

DISCNT=(YES,NF)

Specifies whether VTAM indicates final-use status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES.

This value does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

If you code F or use the default, final-use status is indicated and the connection can be ended. If you code NF, not-final-use status is indicated and the connection should not be ended.

Each device has its own requirements regarding final-use status. To determine whether to code F or NF for a given device, consult the appropriate installation publication for the device.

DLOGMOD

```
►►──┬──────────────────────────────────────────────┬──────────────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name──┘
```

statements:	*LU*
dependencies:	*MODETAB*

Specifies the name of the default logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on logon mode table entries, see “Logon Mode Table” on page 698.

DYNADJCP

```
          ┌─DYNADJCP=value_of_dynadjcp_start_option──────┐
►►──┬──────┼──────────────────────────────────────────────┼──────────────────────────────►◄
    │      │              ┌─value_of_dynadjcp_start_option─┐
    │      └─DYNADJCP=────┼──────────────────────────────┼──┘
    │                     ├─NO───────────────────────────┤
    │                     └─YES──────────────────────────┘
    │  ┌─DYNADJCP=YES─┐  (1)
    └──┼──────────────┼──────────────────────────────────────────────────────────────────┘
       └─DYNADJCP=YES─┘
```

Notes:

1 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *APPN only*

Specifies whether an ADJCP representation of an adjacent control point can be dynamically created for the adjacent node.

DYNADJCP=NO
Indicates that the ADJCP cannot be created dynamically for the adjacent node.

Note: If the TRLE operand is coded, DYNADJCP must be specified as YES or allowed to default. See “TRLE” on page 289 for more information.

DYNADJCP=YES
Indicates that the ADJCP can be created dynamically for the adjacent node.

DYNLU

```
                ┌─DYNLU=value_of_dynlu_start_option──────┐
►►──────────────┼────────────────────────────────────────┼──────────────────────────►◄
                │          ┌─value_of_dynlu_start_option─┐│
                └─DYNLU=───┼─────────────────────────────┼┘
                           ├─NO──────────────────────────┤
                           └─YES─────────────────────────┘
                ┌─DYNLU=YES─┐   (1)
                │           │
                └─DYNLU=YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *CDRDYN*

Specifies whether dynamic allocation of CDRSC definitions is supported. If you do not code this operand, the default is the value you coded on the DYNLU start option.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, a resource that was dynamically defined as a CDRSC cannot use that PU to connect to a session.

Note: If the TRLE operand is coded, DYNLU must be specified as YES or allowed to default. See “TRLE” on page 289 for more information.

DYNLU=NO
Specifies that dynamic allocation of CDRSC definitions is not supported. If you code DYNLU=NO and plan to connect to other nodes using CP-CP sessions, then predefine each node in the CDRSC major node. If you do not, CP-CP session setup fails.

DYNLU=YES
Specifies that dynamic allocation of CDRSC definitions is supported.

For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

Note: If you have coded a DYNLU value in the adjacent CP major node for the adjacent CP to which this physical unit is linked, that value overrides the DYNLU value you code on this PU definition statement. For additional DYNLU considerations, see “DYNLU” on page 21 under the adjacent control point major node.

ENCR

```
►►─┬─ENCR=NONE──────────┬──────────────────────────────────────────►◄
   │                    │
   └─ENCR=─┬─NONE──┬────┘
           │       │
           ├─OPT───┤
           ├─REQD──┤
           └─SEL───┘
```

statements: *LU*
dependencies: *none*

Tells whether this logical unit has any special requirements for enciphering and deciphering messages.

ENCR=NONE
Specifies that this logical unit cannot engage in cryptographic sessions.

ENCR=OPT
Specifies that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from this logical unit. If ENCR=REQD, no sessions can be established with this logical unit unless the host with which it is associated can handle cryptographic sessions.

ENCR=SEL
For a model major node, SEL has no meaning for logical units. If you specify ENCR=SEL, ENCR=OPT is used instead.

ENCRTYPE

```
►►─┬─ENCRTYPE=DES───────┬──────────────────────────────────────────►◄
   │                    │
   └─ENCRTYPE=TDES24────┘
```

statements: *LU*
dependencies: *none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

The SYSDEFed ENCRTYPE of a LU can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to "MODEENT Macroinstruction" on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements:	*LU*
dependencies:	*TSO sessions only*

Determines how sessions are established for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The PLU acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU's application program, only a single BIND is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) that establish sessions with TSO.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application

programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FEATUR2

```
►►─┬─FEATUR2=(DUALCSE,MODEL1,NOEDATS,NOPRINTR,NOSELPEN)─┬──────────────────────────►◄
   │                                                    │
   └─FEATUR2=(──┬──────────────────────────────────────┬──)──────────────────────────┘
                │ ┌─DUALCSE──┐                         │
                │ │          │                         │
                │ └─LOWERCSE─┘                         │
                │ ┌─MODEL1───┐                         │
                │ │          │                         │
                │ └─MODEL2───┘                         │
                │ ┌─NOEDATS──┐                         │
                │ │          │                         │
                │ └─EDATS────┘                         │
                │ ┌─NOPRINTR─┐                         │
                │ │          │                         │
                │ └─PRINTR───┘                         │
                │ ┌─NOSELPEN─┐                         │
                │ │          │                         │
                │ └─SELPEN───┘                         │
```

statements:	*LU*
dependencies:	*none*

Contains certain device information for virtual logical units that are supported by the network terminal option (NTO).

FEATUR2=EDATS

FEATUR2=NOEDATS
Specifies whether this terminal has the extended data stream feature. You cannot use this operand for terminals attached by SDLC lines.

FEATUR2=DUALCSE

FEATUR2=LOWERCSE
Tells VTAM how to send alphabetical characters coded with the TEXT operand on a USSMSG macroinstruction to a non-SNA terminal over the SSCP-LU session. This value does not affect non-alphabetical characters or any characters coded on the BUFFER operand of a USSMSG macroinstruction.

Code LOWERCSE to indicate that alphabetical characters are sent to the terminal over the SSCP-LU session in lowercase. Code DUALCSE to indicate that VTAM sends all characters as they are coded in the USSMSG macroinstruction.

FEATUR2=MODEL1

FEATUR2=MODEL2
Identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

This information is available to an application program as part of the device characteristics pertaining to this terminal. You can obtain those characteristics

by using the INQUIRE macroinstruction. For more information on using the INQUIRE macroinstruction, see *OS/390 eNetwork Communications Server: SNA Programming*.

FEATUR2=NOPRINTR

FEATUR2=PRINTR
Tells VTAM whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only if TERM=3275.

FEATUR2=NOSELPEN

FEATUR2=SELPEN
Specifies whether this terminal supports a selector pen.

INCLUD0E

```
  ┌─INCLUD0E=NO──────┐
►►─┤                  ├──────────────────────────────────────────────►◄
  └─INCLUD0E=─┬─NO─┬─┘
              └─YES┘
```

statements: *PU*
dependencies: *none*

Specifies whether VTAM provides LU names to the logical units defined under this PU. The names are provided by passing a CV0E, Network Name control vector on the ACTLU requests.

INCLUD0E=NO
Indicates that VTAM will not provide LU names to the logical units defined under this PU.

INCLUD0E=YES
Indicates that VTAM will provide LU names to the logical units defined under this PU.

IRETRY

```
  ┌─IRETRY=NO────────┐
►►─┤                  ├──────────────────────────────────────────────►◄
  │            ┌─NO─┐ │
  └─IRETRY=────┤    ├─┘
               └─YES┘
```

statements: *PU*
dependencies: *none*

Determines whether the boundary NCP (the NCP to which the switched physical unit becomes connected) retries a polling operation immediately for the device if an idle-detect timeout follows a polling operation. For more information on this operand, see the *NCP, SSP, and EP Resource Definition Reference*.

Note: If the TRLE operand is coded, IRETRY is ignored. See “TRLE” on page 289 for more information.

LIMRES

```
                ┌─LIMRES=NO──────┐
►►──┬──────────┼────────────────┼──────────────────────────────────────────►◄
    │          │            ┌─NO─┐│
    │          └─LIMRES=────┼────┼┘
    │                       └─YES┘
    │  ┌─LIMRES=NO─┐   (1)
    └──┼───────────┼──────────────────────────────────────────────────────
       └─LIMRES=NO─┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*LIMQSINT; LU 6.2 only; PUTYPE=1 or 2*

Specifies whether this physical unit is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes any sessions that traverse the resource to be deactivated if no conversation is active. A session can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

Note: If the TRLE operand is coded, LIMRES must be specified as NO or allowed to default. See “TRLE” on page 289 for more information.

If you do not code a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

You can also use LIMRES in conjunction with the LIMINTCP start option to specify the interval to retain a free session using a limited resource for some network management flows (excluding flows using the APPN control point sessions or dependent LU server sessions). If you do not code a value for the LIMINTCP start option, resources are not considered to be limited resources, and the sessions will remain active even after all conversations have ended.

LIMRES=NO
Specifies that this physical unit is not to be treated as a limited resource.

LIMRES=YES
Specifies that this physical unit is to be treated as a limited resource.

LOCADDR

```
►►──LOCADDR=address──────────────────────────────────────────────────────►◄
```

statements:	*LU*
dependencies:	*see note*
range:	*0–255 (PU Type 2)* *0–63 (PU Type 1)*

Specifies the logical unit's local address at the physical unit.

For dependent logical units, the range of valid local addresses depends on the PUTYPE of the associated physical unit.

Note: The value of LOCADDR depends upon the requirements of the device being defined. Consult the appropriate component description manual for these restrictions.

LOGAPPL

```
►►──┬──────────────────────────────┬──────────────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu─┘
```

statements: *LU*
dependencies: *NetView or TSO*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Notes:

1. If an error (INOP) occurs for the switched connection, VTAM reallocates the LOGAPPL session and attempts to redial the logical unit to recover the connection. In other words, the session will be terminated by breaking the dial connection instead of by normal session termination using LOGOFF and TERMSELF.

   If you do not code the LOGAPPL operand, the PU and LU must be activated manually.
2. If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────────►◄
    └─LOGTAB=interpret_table_name────┘
```

statements: *LU*

dependencies: *none*

Specifies the name of an interpret table that VTAM uses when processing logon requests originating from the logical unit. VTAM uses the interpret table to interpret the name received from an initiation request. See “Interpret Table” on page 690 for a description of how to define an interpret table.

LUGROUP

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └LUGROUP=lu_group_name─────┘
```

statements: *PU*
dependencies: *none*

Specifies the name of the model LU group that VTAM uses to select a model LU definition when dynamically defining an LU. This operand also indicates that the device supports dynamic definition of LUs.

Note: If the TRLE operand is coded, LUGROUP is ignored. See “TRLE” on page 289 for more information.

LUSEED

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └LUSEED=lu_pattern_name────┘
```

statements: *PU*
dependencies: *LUGROUP*

Provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

This operand is valid only when you code the LUGROUP operand.

Note: If the TRLE operand is coded, LUSEED is ignored. See “TRLE” on page 289 for more information.

If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- The first character must be either alphabetical or one of the national characters (@ or $).
- The remaining characters can be either alphabetical, numerical, or national.
- There must be either two or three contiguous national # characters in the name. These characters represent holders, in which the logical unit’s local address is substituted when an LU’s name is generated. If you specify three # characters, the LU’s logical address is substituted in decimal for the # characters. If you specify two # characters, the LU’s logical address is substituted in hexadecimal for the # characters.

When used with a user-written SDDLU exit routine, this operand is optional and there are no restrictions on its format. For more information on the SDDLU exit, see *OS/390 SecureWay Communications Server: SNA Customization*.

MAXDATA

For PU type 1:

```
►►─┬─MAXDATA=261──────────────────┬──────────►◄
   └─MAXDATA=number_of_bytes──────┘
```

statements:	PU
dependency:	none
range:	5-65535

For PU type 2:

```
►►─┬─MAXDATA=265──────────────────┬──────────►◄
   └─MAXDATA=number_of_bytes──────┘
```

statements:	PU
dependency:	none
range:	5-65535

Specifies the maximum number of bytes that the physical unit can receive in one PIU or PIU segment, including the transmission header (TH) and request or response header (RH). To determine the maximum PIU (or PIU segment) size that the physical unit can receive, consult the component publications for the specific type of SDLC station represented by this PU definition statement.

The maximum amount of user data that VTAM sends to the physical unit in one PIU is:
- MAXDATA minus 2 bytes (for a PU type 1)
- MAXDATA minus 6 bytes (for a PU type 2)

Note: The RU part of the PIU contains the user data. When VTAM segments the PIU, the middle and last PIUs contain three more bytes of user data in the RU because there is no RH field. The value for the PIU varies with the device; for example, the 3276 requires MAXDATA=262.

MAXDATA is ignored for type 2.1 nodes attached through an NCP with peripheral node type 2.1 support. The type 2.1 node supplies the actual value used when the connection is established.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information about physical units connected through an NCP.

Note: If the TRLE operand is coded, MAXDATA is ignored. See “TRLE” on page 289 for more information.

Notes:

1. In a local area network, the value for the MAXDATA operand on the PU definition statement must be less than or equal to the value for the MAXDATA operand on the PORT definition statement for the LAN major node to which the switched physical unit is connected.
2. NCP V4R3 changed the MAXDATA calculation for a type 1 physical unit. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on calculating a MAXDATA value for physical units on nonswitched links.

MAXOUT

```
   ┌─MAXOUT=1──────────────────┐
►►─┤                           ├──────────────────────────────────────────►◄
   └─MAXOUT=max_number_of_pius─┘
```

statements:	*PU*
dependencies:	*none*
range:	*1–127*

Specifies the maximum number of PIUs or PIU segments (if the session parameters allow segmenting of data) that VTAM sends to the physical unit represented by this definition statement before requesting a response.

For performance reasons, you should code MAXOUT=7 if the physical unit represented by this definition statement can handle more than one PIU before sending a link-level response. To determine this, consult the component publications for the specific type of SDLC station.

Note: If the TRLE operand is coded, MAXOUT is ignored. See “TRLE” on page 289 for more information.

MAXSESS

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─MAXSESS=max_number_LU-LU_sessions─┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*independent LUs attached to an NCP*
range:	*1–65535*

Specifies the maximum number of concurrent LU-LU sessions in which an independent LU can participate. The MAXSESS value applies to each connection the LU is using to establish LU-LU sessions. For example, if MAXSESS=2 was specified for LUA and LUA appears on three PUs, LUA can have two sessions on each of the three PUs for a total of six sessions.

If you do not code MAXSESS or code a value outside the range, the MAXSESS value defined on the BUILD definition statement (in the NCP major node) will be used. Reference the *NCP, SSP, and EP Resource Definition Reference* for more information about this operand.

Note: If the TRLE operand is coded, MAXSESS is ignored. See “TRLE” on page 289 for more information.

MDLENT

```
►►──┬──────────────────────────────────┬──────────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name─┘
```

statements:	*LU*

dependencies: *MDLTAB*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─┘
```

statements: *LU*
dependencies: *none*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

If you omit the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MODETAB

```
    ┌─MODETAB=ISTINCLM──────────────┐
►►──┼──────────────────────────────┼──────────────────────────────────────►◄
    └─MODETAB=logon_mode_table_name─┘
```

statements: *LU*
dependencies: *none*

Specifies the name of a logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►──┬──────┬──────────────────────────────►◄   ►►──name──────────────────────────────►◄
    └name──┘
```

statements: VBUILD *statements:* PU, LU

Provides the name associated with the resource being defined.

On the VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On the PU: Assigns the required minor node name of a model physical unit that is used in conjunction with the installation exit routine to build a dynamic switched definition.

For a model PU for XCF connectivity, *name* can contain up to five characters. The first 1–4 characters determine the prefix for the PU names added to the dynamic local SNA major node (ISTLSXCF). The last character must be an asterisk (*).

On the LU: Assigns the required minor node name of the model logical unit that is used in conjunction with the installation exit routine to build a dynamic switched definition.

NATIVE

```
►►──┬──────────────────┬──────────────────────────────────────────────────────────►◄
    └NATIVE=──┬──NO──┬──┘
              └──YES─┘
```

statements: *PU*
dependencies: *BN=YES; APPN only*

Specifies whether this link station represents a connection to a native node.

Note: If the TRLE operand is coded, NATIVE is ignored. See "TRLE" on page 289 for more information.

If NATIVE is not specified on either the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

No subnetwork negotiations will occur if NATIVE has been specified; in this case the value between the two nodes must match for connection to occur.

If you code NATIVE on both the ADJCP and PU definition statements, the values must match. If a transmission group (TG) is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the TG attempting activation is not allowed to become active. If the activation is for a virtual-route-based transmission group then the SSCP-SSCP session will also fail.

NETID

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─NETID=network_id─────────────┘
```

statements:	*PU*
dependencies:	*XNETALS start option*

Specifies a 1–8 character network name.

NETID is used in conjunction with the XNETALS start option in support of nonnative network connection. See page “XNETALS” on page 657 for information about XNETALS.

Notes:

1. The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.
2. If the TRLE operand is coded, NETID is ignored. See “TRLE” on page 289 for more information.

NETID=*network_id*
Specifies that VTAM ensures that the connecting resource is within the network specified by NETID when the physical unit is active. If you omit NETID, VTAM dynamically acknowledges the network ID during connection establishment. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM’s network ID.

If dynamic dial-out connections are required (that is, a session request drives the dial), NETID must be predefined. Otherwise, VTAM might not be able to determine the connecting network ID and the resultant sessions might fail. If leased connections are not established, and if NETID is not predefined, directory requests could fail because of VTAM’s inability to determine the resource’s actual network.

NN

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─NN=──┬─NO──────────┬──┘
           └─YES─────────┘
```

statements:	*PU*
dependencies:	*APPN only*

Specifies whether the adjacent node is expected to be a network node.

- If a value for the NN operand is specified, it will be validated when a connection is attempted to the adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails.
- If the NN operand is not specified, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

Note: If the TRLE operand is coded, NN is ignored. See “TRLE” on page 289 for more information.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

PACING

```
►►─┬─PACING=1──────────────────┬──────────────────────────────────────────►◄
   │                           │
   └─PACING=─┬─1──────────────┬─┘
             └─pacing_number──┘
```

statements:	*LU*
dependencies:	*none*
range:	*0–63*

Specifies how pacing is handled between the logical unit and the boundary node to which it is connected.

Note: You can also use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and used fixed pacing instead. To use the FIXED option, you must code (N,,FIXED). The second comma represents a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

PASSLIM

```
►►─┬─PASSLIM=1──────────────┬──────────────────────────────────────────►◄
   │                        │
   └─PASSLIM=number_of_pius─┘
```

statements:	PU
dependencies:	none
range:	1–value of MAXOUT

Determines, for switched physical units connected through an NCP, the maximum number of contiguous PIUs that the NCP sends to the physical unit at one time.

Note: If the TRLE operand is coded, PASSLIM is ignored. See “TRLE” on page 289 for more information.

PDELAY

```
   ┌─PDELAY=TERRESTR──────┐
   │                (1)   │
   ├─PDELAY=NEGLIGIB──────┤
►►─┼──────────────────────┼──────────────────────────────────────────►◄
   │          ┌─TERRESTR─┐ │
   └─PDELAY=──┼──────────┼─┘
              ├─LONG─────┤
              ├─NEGLIGIB─┤
              └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

statements:	*PU*
dependencies:	*APPN only*

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

See “TRLE” on page 289 for more information about coding PDELAY for XCF connectivity.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PUTYPE

```
   ┌─PUTYPE=2─────────┐
►►─┼──────────────────┼──────────────────────────────────────────►◄
   │          ┌─2─┐   │
   └─PUTYPE=──┼───┼───┘
              └─1─┘
   ┌─PUTYPE=2─┐   (1)
   │          │
   └─PUTYPE=2─┘
```

Notes:

1 PUTYPE=2 is the only valid value if the TRLE operand is coded.

statements:	*PU*
dependencies:	*none*

Specifies the physical unit type. The physical unit type depends on the type of cluster controller (such as 3767, 3174, or AS/400®*). To determine the physical unit type for a given device, see the component description manual for that device.

Note: If the TRLE operand is coded, PUTYPE must be specified as 2 or allowed to default. See “TRLE” on page 289 for more information.

For a type 2.1 physical unit, code PUTYPE=2.

The physical unit type for a LAN switched station must be 2.

For PU types 1 and 2, or type 2.1, the value of PUTYPE occupies bits 4–7 of the 48-bit station ID constructed by VTAM for switched network operation. For a description of the station ID, see “IDNUM” on page 437.

REGISTER

For Independent LUs:

```
►►─┬─REGISTER=NO──────────────┬──────────────────────────►◄
   │                          │
   └─REGISTER=─┬─NO──────────┬┘
               ├─CDSERVR─────┤
               └─NETSRVR─────┘
```

statements:	LU
dependency:	none

For Dependent LUs:

```
►►─┬─REGISTER=NETSRVR─────────┬──────────────────────────►◄
   │                          │
   └─REGISTER=─┬─NETSRVR─────┬┘
               ├─CDSERVR─────┤
               └─NO──────────┘
```

statements:	LU
dependency:	none

Specifies how a resource should be registered. The default depends on whether you code REGISTER for dependent or independent LUs.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

REGISTER=NO
Indicates that the resource should not be registered.

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

RESETLU

```
►►─┬─RESETLU=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─RESETLU=─┬─YES─┬─────┘
              └─NO──┘
```

statements: *PU, LU*
dependencies: *dependent LUs only*

Specifies whether a DACTLU request followed by a ACTLU request will be sent during clean-up processing.

RESETLU=YES
Specifies that during clean-up processing for dependent LUs, VTAM will issue an ACTLU request after the DACTLU request.

RESETLU=NO
Specifies that during clean-up processing for LU-LU sessions, RESETLU=NO prevents a DACTLU-ACTLU from flowing to an LU.

SECNET

```
►►─┬─SECNET=NO────────────┬──────────────────────────────────────────►◄
   │                      │
   │                      │
   └─SECNET=─┬─NO──┬─────┘
             └─YES─┘
   ┌─SECNET=NO──┐   (1)
   │            │
   └─SECNET=NO──┘
```

Notes:

1 SECNET=NO is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *see description*

Specifies whether this physical unit is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

This operand applies only if you have IBM 3710 Network Controllers or IBM 3174 token-ring attachments in your network.

SECNET=NO
Specifies that this physical unit is not associated with the secondary network.

SECNET=YES
Specifies that this physical unit is associated with the secondary network. When SECNET=YES, data that is received by VTAM from this physical unit is flagged as requiring special problem-determination considerations when routed to a communication network management application.

Note: If the TRLE operand is coded, SECNET must be specified as NO or allowed to default. See “TRLE” on page 289 for more information.

SECURITY

```
   ┌─SECURITY=UNSECURE──────┐
►►─┼────────────────────────┼──────────────────────────────────────────────►◄
   │                ┌─UNSECURE─┐
   └─SECURITY=──────┼──────────┼─┘
                    ├─PUBLIC───┤
                    ├─UNDERGRO─┤
                    ├─SECURE───┤
                    ├─GUARDED──┤
                    ├─ENCRYPT──┤
                    └─SHIELDED─┘
```

statements:	*PU*
dependencies:	*APPN only*

Specifies the security level of the transmission group.

See “TRLE” on page 289 for more information about coding SECURITY for XCF connectivity. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SPAN

```
►►─┼──────────────────┼──────────────────────────────────────────────────►◄
   └─SPAN=span_name───┘
```

statements:	*PU,LU*
dependencies:	*NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *NetView Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SSCPFM

```
►►─┬─SSCPFM=FSS──────────┬──────────────────────────────────────────►◄
   │                     │
   │           ┌─FSS─────┐
   └─SSCPFM=───┼─────────┼─┘
               ├─USSNTO──┤
               ├─USSSCS──┤
               ├─USS3270─┤
               ├─USS3275─┤
               └─USS3780─┘
```

statements: *LU*
dependencies: *none*

Determines what type of RUs the logical unit can support in its communications with the SSCP.

SSCPFM=FSS
Specifies that field-formatted RUs are supported.

SSCPFM=USSNTO
Specifies that this logical unit is supported by Network Terminal Option (NTO) in the boundary node, and character-coded RUs are supported over its SSCP-LU sessions.

SSCPFM=USSSCS
Specifies that character-coded RUs are supported.

SSCPFM=USS3270
Allows printing of a user-defined logon message (USSMSG10) on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

SSCPFM=USS3275
Prevents printing of a user-defined logon message (USSMSG10) for a terminal defined as an SDLC 3275 (PU type 1) that might have an attached printer. Code SSCPFM=USS3275 to prevent the user-defined logon message from printing on BSC printers.

SSCPFM=USS3780
Coded for non-SNA devices supported with NTO that use the 3780 protocol.

These devices use USS command facilities such as LOGON and IBMTEST and are supported only by Release 2 of NTO. Refer to the *NTO Installation* for more information.

TERM

```
►►─┬──────────────────────────┬──────────────────────────────────────►◄
   │                          │
   └─TERM=──┬─3275────────────┬┘
            ├─3277────────────┤
            ├─3284────────────┤
            ├─3286────────────┤
            └─terminal_Type───┘
```

statements: *LU*

dependencies: *none*

Identifies, for a virtual logical unit supported through NTO, the device data stream compatible characteristics. An application program can find these characteristics by executing an INQUIRE DEVCHAR macroinstruction that names this virtual logical unit. The information is placed in the DEVAUXTP field of the DEVCHAR DSECT. (DEVAUXTP is an 8-byte storage area provided by the application program.) For more information on the DEVCHAR DSECT, see *OS/390 eNetwork Communications Server: SNA Programming*.

TGP

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─TGP=tgp_name─────┘
```

statements: *PU*
dependencies: *APPN only*

Specifies the name of a transmission group (TG) profile. The characteristics of the TG profile (along with any modifiers on this statement) become the characteristics of the PU. If TGP is not specified or has not been activated when the PU becomes active, default TG characteristic are assigned.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TOPO

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─TOPO=CONNECT─────┘
```

statements: *PU*
dependencies: *APPN only*

Controls the way APPN connections using this link station are reported to APPN topology and routing services for inclusion into APPN functions.

Note: Except in the switched major node, the only value for TOPO is CONNECT. If you specify APPN on the CONNTYPE start option, it is not necessary to code TOPO, as you automatically get a value of CONNECT.

TOPO=CONNECT
Specifies that APPN topology and routing services receives information about the connection and its characteristics when the connection is activated. When the connection is deactivated, APPN topology and routing services receives notification that the TG is inactive.

TRLE

```
►►──┬──────────────────────┬──────────────────────────────────────►◄
    └─TRLE=──┬─trle_name─┬──┘
             └─*─────────┘
```

statements: *PU*
dependency: *XCF connectivity only*

Specifies that this model PU is for XCF connectivity.

Note: A model for dynamic XCF local SNA PUs can be defined to override the XCF default values for the operands of the PU definition statement.

The *trle_name* can contain up to five characters. The first 1–4 characters determine the prefix for the TRLE names added to the TRL major node (ISTTRL). The last character must be an asterisk (*). If you code a single asterisk, the default prefix of ISTT* is used for the TRLEs added to the TRL major node (ISTTRL).

The requirements for the *name* of an XCF model PU and the *trle_name* are different from the requirements for names of other resources. These requirements are:

- To use the default prefix of ISTP* for dynamic XCF local SNA PUs or ISTT* for the TRLEs added to the TRL major node, you can specify a single asterisk (*).
- To specify a different prefix, the following rules apply:
  - The name specified must be from 2–5 characters.
  - The first character can be alphabetical (A–Z) or the national characters @, #, or $.
  - The last character must be an asterisk (*).
  - Any characters between the first and last characters can be alphabetical (A–Z), numerical (0–9), or the national characters @, #, or $.

Refer to Table 30 for the operands and values that can be specified when TRLE is coded.

Table 30. XCF Model PU Operands Used With TRLE

Operand	Coding Required?	Value Required for XCF	Default for XCF
name	Yes	User specified value	None
CAPACITY	No	User specified value	8K
CONNTYPE	No	APPN	APPN *
COSTBYTE	No	User specified value	0
COSTTIME	No	User specified value	0
CPCP	No	User specified value	Value of CPCP start option
DISCNT	No	User specified value	NO
DYNADJCP	No	YES	YES *
DYNLU	No	YES	YES *
LIMRES	No	NO	NO
PDELAY	No	User specified value	NEGLIGIB *
PUTYPE	No	2	2
SECNET	No	NO	NO
SECURITY	No	User specified value	SHIELDED *
SPAN	No	User specified value	None
TGP	No	User specified value	None

Table 30. XCF Model PU Operands Used With TRLE (continued)

Operand	Coding Required?	Value Required for XCF	Default for XCF
TOPO	No	User specified value	CONNECT
UPARM1	No	User specified value	128
UPARM2	No	User specified value	128
UPARM3	No	User specified value	128
VERALSID	No	NO	NO
* The XCF default for this operand is different than the default for other model PUs.			

TYPE

```
►►──TYPE=MODEL──────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Means that this VBUILD definition statement defines a model major node to VTAM. Dynamic switched definitions for previously undefined devices will be built using the PU and LU definitions contained within this node.

UPARM1

```
                ┌─UPARM1=128──────────────────────┐
►►──────────────┼─────────────────────────────────┼──────────────►◄
                └─UPARM1=user-defined_operand_value─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
                ┌─UPARM2=128──────────────────────┐
►►──────────────┼─────────────────────────────────┼──────────────►◄
                └─UPARM2=user-defined_operand_value─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM3=user-defined_operand_value─┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

USSTAB

```
►►─┬─USSTAB=ISTINCDT──────┬──────────────────────────────────────────────►◄
   └─USSTAB=uss_table_name─┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

A terminal user can issue a USS command with the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

USSTAB is valid for dependent logical units only.

VERALSID

```
      ┌─VERALSID=NO──────────┐
►►────┼──────────────────────┼──────────────────────────────────────────────►◄
      │                ┌─NO─┐│
      └─VERALSID=──────┼────┼┘
                       └─YES┘
      ┌─VERALSID=NO──┐   (1)
      │              │
      └─VERALSID=NO──┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

statements: *PU*
dependencies: *APPN only*

Specifies whether the adjacent link station name should be used to determine the link station. When specified as YES, the connection endpoints must supply the

same name in the CV0EF7 vector that is appended to XID3s. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

Note: If the TRLE operand is coded, VERALSID must be specified as NO or allowed to default. See “TRLE” on page 289 for more information.

VPACING

```
►►─┬─VPACING=2──────────────────────┬──────────────────────────────────────►◄
   │                                │
   └─VPACING=─┬─2──────────────────┬┘
              └─number_of_requests─┘
```

statements:	*LU*
dependencies:	*none*
range:	*0–63*

Determines how VTAM paces the flow of data from VTAM to the boundary node to which the logical unit is connected.

Note: You can also use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and used fixed pacing instead. To use the FIXED option, you must code (N,,FIXED). The second comma represents a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

VTAMTOPO

```
►►──┬──────────────────────────────┬──►◄
    │                              │
    └─VTAMTOPO=─┬─REPORT───┬───────┘
                └─NOREPORT─┘
```

statements:	PU
dependency:	none

Specifies whether the VTAM topology agent reports the dynamic PUs created from PU model definitions to the CMIP manager applocation programs for SNA local topology. This will affect only dynamic PUs created after any changes to the model definition. It will not affect existing dynamic PUs.

An explicit specification of VTAMTOPO on a model PU definition is used to initialize a dynamic PU created from the model definition, as though the VTAMTOPO value had been coded on the dynamic PU definition. If no VTAMTOPO value is specified on the model PU definition, the reporting state is taken from the default for the

ISTDSWMN major node (not connected) or the related NCP or XCA major node (when the dynamic PU is connected). Consequently, coding VTAMTOPO or VBUILD has no affect at this time.

Specifying VTM.VTAMTOPO allows VTAMTOPO to be handled appropriately by the NCP Definition Facility for versions of NDF that do not recognize VTAMTOPO natively.

VTAMTOPO=REPORT
PU

Specifies that the VTAM topology agent reports this PU.

VTAMTOPO=NOREPORT
PU

Specifies that the VTAM topology agent does not report this PU.

XNETALS

```
     ┌XNETALS=value_of_XNETALS_start_option┐
►►───┤                                     ├──────────────────────────────────►◄
     └XNETALS=──┬─NO──┬──────────────────┘
                └─YES─┘
```

statements:	*PU*
dependencies:	*none*

Determines whether the network ID of an attaching node or VTAM's network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true. If the NETID that the PU sent in its XID is not the same as VTAM's and XNETALS=NO or is defaulted, IST680I will be issued.

XNETALS=NO
Indicates that VTAM's network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement, then it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement, then VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES
Indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Note: Providing that XNETALS=YES is coded, you can prohibit nonnative network attachment for LU-LU sessions for a particular PU by coding XNETALS=NO on the GROUP, LINE or PU statements.

Network Control Program (NCP) Major Node

To define an NCP major node, code the VTAM and NCP definition statements detailed in this section, and file them in the appropriate VTAM definition library. The name under which the major node is filed must match the NCP load module name on the NEWNAME operand of the BUILD definition statement.

Information required by VTAM to communicate with the NCP and its attached devices is provided to VTAM by coding VTAM definition statements and NCP definition statements with VTAM operands. The VTAM definition statements and operands must be coded in the NCP major node, even though they provide no information to the NCP.

Definition statements in the NCP major node can be divided into the following categories:

VTAM only
Statement is processed by VTAM only.

VTAM and NCP
Statement contains operands that are processed by both VTAM and by the NCP generation process.

If there are any VTAM-only operands or restrictions on the statement, you will find a description in this book, but the description of the whole definition statement is in the *NCP, SSP, and EP Resource Definition Reference*.

VTAM and NCP, no VTAM requirements
Statement contains operands that are processed by both VTAM and by the

NCP generation process, but it has no special VTAM requirements and should be coded as described in the *NCP, SSP, and EP Resource Definition Reference*.

NCP only, with VTAM requirements
Statement contains operands processed only by NCP, however there are special VTAM requirements.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for a description and VTAM requirements.

The following list shows the definition statements used by VTAM in node.

PCCU	VTAM only
BUILD	VTAM and NCP
HOST	VTAM only (requires NCP V4R3 or later)
SYSCNTRL	VTAM and NCP
LUDRPOOL	NCP only, with VTAM requirements
PUDRPOOL	NCP only, with VTAM requirements
PATH	VTAM and NCP
GROUP	VTAM and NCP
LINE	VTAM and NCP
CLUSTER	VTAM and NCP
TERMINAL	VTAM and NCP
GROUP	VTAM and NCP
LINE	VTAM and NCP
PU	VTAM and NCP
LU	VTAM and NCP
NETWORK	VTAM and NCP
FRSESET	VTAM and NCP
GWNAU	NCP only, with VTAM requirements
GENEND	VTAM and NCP, no VTAM requirements

For a complete list of definition statements coded in the NCP major node and coding sequence information, reference the *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide*.

Identifying the Communication Controller

The programmed communication control unit (PCCU) definition statement identifies the communication controller where the NCP is loaded. The PCCU definition statement also contains information related to VTAM's support of the NCP.

PCCU Definition Statement

Code at least one PCCU definition statement for each VTAM that is to activate the NCP. It is required in VTAM systems and defines the VTAM functions that are provided for this NCP.

If more than one VTAM host will activate the NCP, you can code a PCCU definition statement for each host. Each PCCU definition statement describes the functions performed by the host it is associated with.

The SUBAREA operand (see "SUBAREA" on page 379) and the NETID operand (see "NETID" on page 364) associate the communication controller with a VTAM host when the NCP is activated. If each host provides the same function for the NCP, only one PCCU definition statement is necessary, but the operator might need to override unique attachment requirements.

PCCU definition statements must be at the beginning of the NCP generation deck.

HOST Definition Statement

Code the HOST definition statement only for NCPs in channel-attached controllers. The following operands provide information that VTAM uses in communicating with the NCP over the channel:
- INBFRS
- MAXBFRU
- NETID
- SUBAREA

A channel-attached NCP requires one HOST definition statement for each host that activates the NCP. Do not define a HOST definition statement for a link-attached NCP. For more information on defining NCP channel support, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

In an SNA-interconnected network, more than one HOST definition statement can exist in the NCP major node. VTAM uses the following steps to determine which HOST definition statement applies to the VTAM host and should be processed:
1. The VTAM host processes the first HOST definition statement whose SUBAREA operand matches VTAM's and whose NETID matches VTAM's.
2. If HOST definition statements as described in step 1 do not exist, the VTAM host processes the first HOST definition statement that has the same SUBAREA value as VTAM and that does not have a NETID specified.

   Note: A match is made on the SUBAREA when the value coded for the SUBAREA operand matches the value of VTAM's SUBAREA or when the SUBAREA operand is not coded and VTAM's subarea is 1.

If parallel TGs are used, only one HOST definition statement is allowed for one host subarea destination. You cannot have different HOST definition statements for every TG.

NCP Definition Statements for the NCP

The following NCP definition statements used to define an NCP load module also contain VTAM-related information:
- BUILD
- SYSCNTRL
- HOST
- LUDRPOOL
- PUDRPOOL
- PATH
- NETWORK
- GWNAU
- GENEND

BUILD Definition Statement

Most of the operands on the BUILD definition statement relate only to the NCP. However, VTAM processes the following operands:
- LOADLIB
- MAXSUBA
- NETID
- NEWNAME

- PUNAME
- SALIMIT
- SUBAREA

SYSCNTRL Definition Statement

SYSCNTRL identifies the dynamic control facilities included in the NCP. These facilities allow the NCP to execute requests from VTAM to change certain NCP parameters or to determine the status of resources such as lines and stations.

VTAM always needs these OPTION parameters for BSC 3270s:
MODE Set destination mode
RIMM Reset immediate

VTAM needs these OPTION parameters for the listed operator commands:

OPTION	VTAM Command	Description
NAKLIM	MODIFY NEGPOLL	Change line negative polling response limit
SESSION	MODIFY SESSION	Change session limit
SSPAUSE	MODIFY POLL	Change service-seeking pause
STORDSP	MODIFY NCPSTOR	Display NCP storage

LUDRPOOL and PUDRPOOL Definition Statements

VTAM support for switched terminals, dynamic reconfiguration, and primary addresses for independent logical units requires that the LUDRPOOL and PUDRPOOL definition statements be coded at NCP generation. To support these functions, VTAM issues request network address assignment (RNAA) RUs requesting the NCP to assign a physical unit or logical unit address to a resource specified by VTAM. The addresses come from the address pools generated as a result of the LUDRPOOL or PUDRPOOL definition statements. For more information on the LUDRPOOL and PUDRPOOL definition statements, refer to the *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide*.

PATH Definition Statements

PATH definition statements are required in the NCP major node to routes to the VTAM subarea node. The PATH statement specifies the explicit routes, adjacent subareas, and transmission groups used to send data to VTAM and other destination subareas. Also, data flow threshold values for the transmission groups are specified. For more information on the PATH definition statement for NCP, refer to the *NCP, SSP, and EP Resource Definition Reference* and the *NCP, SSP, and EP Resource Definition Guide*.

NETWORK Definition Statement (SNA Network Interconnection Only)

You can specify a number of predefined networks in the NCP major node to identify each network attached to a gateway NCP. Those networks are defined using the NETID operand on the BUILD and NETWORK definition statements. You do not have to predefine all networks if you define a model NETWORK definition statement. To define a model network, code the COPIES operand instead of the NETID operand. You must code NETID or COPIES, but not both.

Code only one model NETWORK definition statement for each NCP.

The following operands are processed by VTAM:
- COPIES
- MAXSUBA
- NETID
- SALIMIT
- SUBAREA

FRSESET Definition Statement

The FRSESET (frame-relay switching equipment set) definition statement contains the pairings of the frame relay PUs for a frame relay line.

In a FRSESET statement, you specify the primary frame relay PU partners and their optional backups in the SUBPORTS operand. There are a minimum of two and a maximum of four PUs in a FRSESET definition statement. The FRELAY operand on the GROUP definition statement indicates the type of frame relay group.

The following FRSESET rules are validated by VTAM:
- The name on the FRSESET is required and unique in a domain. If not, a message is issued stating that a duplicate resource has been defined.
- Each puname represents a PU type 1. If not, a message is issued stating the puname is defined incorrectly.
- Each puname listed in SUBPORTS was defined in this NCP source prior to this FRSESET. If the puname has not been defined previously, a message is issued stating an undefined resource has been used in the FRSESET statement.
- The required punames are coded. If not, a message is issued stating that a required parameter has been specified on the FRSESET statement.
- Each puname coded on the SUBPORTS operand must be distinct within SUBPORTS and across FRSESET definition statements. If punames in SUBPORTS are not distinct, a message is issued stating a duplicate resource has been used. If a puname has been used in more than one FRSESET, a message is issued stating that a duplicate resource has been used.
- None of the punames in SUBPORTS is owned by different VTAMs. If so, a message is issued stating that the resource is not owned by this VTAM.
- None of the PUs in SUBPORTS is the first PU under a LINE. The first PU under a LINE is the local management interface (LMI). If a PU in SUBPORTS is the LMI, a message is issued stating that a resource used in the definition is not valid.
- Each puname in the source is in a FRSESET. If the puname is defined as a frame-relay PU and is not in a FRSESET, a message is issued identifying that the PU cannot be processed.

The following rule is not validated by VTAM, but is validated by NCP, which sends a sense code if the rule is not followed:
- Each puname in SUBPORTS is on a distinct frame-relay line.

GWNAU Definition Statement (SNA Network Interconnection Only)

The GWNAU definition statement is an NCP-only definition statement that defines representations of cross-network logical units and SSCPs within the gateway NCP. VTAM does not use the definition statement when processing the NCP generation definitions. In some configurations, VTAM's ability to activate SSCP-SSCP sessions depends on proper use of the GWNAU definition statement. Refer to the *OS/390*

SecureWay Communications Server: SNA Network Implementation Guide for information on how this definition statement relates to session activation.

GENEND Definition Statement

The GENEND definition statement is an NCP-only statement used to delimit the end of the NCP definition statements.

NCP Definition Statements for Resources Controlled by the NCP

The following NCP definition statements are used to define lines, PU type 4 link stations, and peripheral devices to VTAM.

- GROUP
- LINE
- PU
- LU
- CLUSTER
- TERMINAL

Most of the operands on the above definition statements relate to the NCP only. However, there are some that are VTAM only or that have restrictions placed by VTAM. This book contains information about operands processed by VTAM. Use this book in conjunction with the NCP coding requirements described in the *NCP, SSP, and EP Resource Definition Reference*.

GROUP Definition Statement

The GROUP definition statement begins a series of definition statements that define links and devices attached to the NCP with common characteristics.

The following are NCP-only operands that should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding these operands.

- CHANLNK
- LINEADD
- NPARSC
- VIRTUAL

LINE Definition Statement

The LINE definition statement identifies characteristics and functions associated with links (SDLC, BSC, start-stop, 370 I/O channel adapter connections, or channel) connecting the communication controller to another node.

The following are NCP-only operands that should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding these operands.

- ADDRESS
- CUTYPE
- DIRECTN
- PAUSE
- POLLED
- SESSION
- TYPE

PU Definition Statement

The PU definition statement defines SDLC devices, channel adapter connections, and adjacent NCP connections to the NCP. The PU specifies functions and characteristics related to these connections.

The following are NCP-only operands that should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding these operands.
- BNNSUP
- IRETRY
- MAXDATA
- MAXLU
- MAXOUT
- PASSLIM

LU Definition Statement

The LU definition statement specifies characteristics and functions for the logical units associated with a device.

The following NCP-only operand should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding this operand.
- BATCH

CLUSTER Definition Statement

The CLUSTER definition statement defines clustered BSC stations attached to the NCP.

The following are NCP-only operands that should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding these operands.
- CUTYPE
- PASSLIM

TERMINAL Definition Statement

The TERMINAL definition statement defines BSC and start-stop devices attached to the NCP.

The following are NCP-only operands that should be looked at carefully for VTAM restrictions. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on coding these operands.
- ADDR
- CONV
- DIRECTN

Table 31. NCP Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PCCU	AUTODMP	324			NO	DUMPDS
		AUTOIPL	325			NO	
		AUTOSYN	325			YES	
		BACKUP	326			NO	OWNER

Table 31. NCP Major Node Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		CDUMPDS	329			DUMPDS value	
		CHANCON	329			COND	
		CONFGDS	331				
		CONFGPW	331				CONFGDS
		CUADDR	334				
		DELAY	335			0	
		DUMPDS	338				
		DUMPLD	338			NO	
		DUMPSTA	339				
		GWCTL	348			SHR	SNI only
		INITEST	349			NO	3705 only
		LOADFROM	354			HOST	see description
		LOADSTA	355				
		MAXDATA	360			65535	
		MDUMPDS	362			DUMPDS value	
		NETID	364				see description
		NEWPATH	365				
		OWNER	366				VTAM
		RNAME	374				
		SAVEMOD	375				3720 or 3745
		SUBAREA	379				
		TGN	381			1 (Subarea) ANY (APPN)	
		VFYC	386			YES	
		VFYLM	386			NO	
name	BUILD	LOADLIB	355	R			
		NETID	229				XNETALS
		SUBAREA	379	R			VTAM & NCP
		COSTAB	333			ISTSDCOS	
		MAXSUBA	361				Required if PRE-ENA node in network VTAM & NCP
		NEWNAME	365				
		PUNAME	371				
		SALIMIT	375			255	
		VTAMTOPO	389			REPORT	DIAL=YES
name	SYSCNTRL	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					
name	HOST	INBFRS	349	R			
		MAXBFRU	359	R			
		NETID	364				
		SUBAREA	379			1	
name	LUDRPOOL	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					
name	PUDRPOOL	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					

Table 31. NCP Major Node Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	NETWORK	NETID	364	R			SNI only; COPIES
		SUBAREA	379	R			SNI only
		COPIES	332				SNI only; NETID
		COSTAB	333			ISTSDCOS	SNI only
		MAXSUBA	361				For pre-ENA; SNI only; VTAM & NCP
		SALIMIT	375			255	SNI only
name	FRSESET	SUBPORTS	380	R			VTAM & NCP; PU type 1
name	GWNAU	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					
name	GROUP	LNCTL	353	R			VTAM & NCP
		DIAL	336			NO	VTAM & NCP
		DYNPU	341			NO	
		DYNPUPFX	342			CN	DYNPU=YES
		ECLTYPE	342				VTAM & NCP
		ENCR	343		G,L,P	NONE	VTAM
		ENCRTYPE	343		G,L,P	DES	
		FRELAY	346				VTAM & NCP; PU type 1
		PHYSRSC	369				
		PUTYPE	371			5	VTAM & NCP
		SHM	377			NO	VTAM
		VTAMTOPO	389				DIAL=YES
name	LINE	ANSWER	322		G	ON	VTAM
		AUTO	323		G		VTAM & NCP
		AUTODL	324		G	YES	VTAM & NCP
		BUFCAP	327		G	1	VTAM
		CALL	327		G	IN	VTAM & NCP
		CKEYNAME	330			YES	VTAM
		ENCR	343		G,L,P	NONE	VTAM
		ENCRTYPE	343		G,L,P	DES	
		ISTATUS	351		G	ACTIVE	VTAM
		LINEAUT	352		G		VTAM & NCP
		LOCADD	355				VTAM & NCP
		POLIMIT	370		G	(1,NOWAIT)	VTAM & NCP
		PU	370		G	YES	VTAM
		SWNORDER	380		G	CPNAME	VTAM
		TERM	381		G		VTAM & NCP
		USE	384		G		VTAM & NCP USE=REDEF USE=SPARE
		VNGROUP	387		G		VNNAME; VTAM only ECLTYPE=PHYSICAL
		VNNAME	387		G		VNGROUP; VTAM only ECLTYPE=PHYSICAL
		VTAMTOPO	389		G		DIAL=YES

Table 31. NCP Major Node Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PU	ADDR	321				VTAM & NCP; DLCI
		ANS	321		G,L	STOP	VTAM & NCP
		AUTHLEN	323		G,L	YES	Type 2.1
		CAPACITY	328		G,L	8K or 4M	APPN only
		CONNTYPE	331		G,L	CONNTYPE start option	APPN only
		COSTBYTE	333		G,L	0	APPN only
		COSTTIME	333		G,L	0	APPN only
		CPCP	334		G,L	CPCP start option	APPN only
		CPNAME	334				
		DATMODE	335		G,L	HALF	VTAM & NCP
		DISCNT	336		G,L	(NO)	VTAM
		DLCI	337				VTAM & NCP; ADDR
		DYNADJCP	340		G,L	DYNADJCP start option	APPN only CPNAME
		DYNLU	340		G,L	DYNLU start option	PUTYPE=1, 2; VTAM
		ENCR	343		G,L,P	NONE	VTAM
		ENCRTYPE	343		G,L,P	DES	
		GP3174	347		G,L	NO	VTAM & NCP
		HPR	348		G,L	YES	VTAM; APPN only; NCP V7R3
		INCLUD0E	349			NO	
		INNPORT	350			NO	VTAM
		ISTATUS	351		G	ACTIVE	VTAM
		LIMRES	351		G,L	NO	VTAM; LIMQSINT; LU 6.2; PUTYPE=1, 2
		LLERP	352			see description	VTAM; HPR=YES
		LMODADR	353			see description	VTAM & NCP LPDATS; SEGMENT
		LPDA®	357			ALLOW	VTAM & NCP
		LUGROUP	358				VTAM
		LUSEED	359				VTAM
		NATIVE	363		G,L		BN=YES APPN only
		NETID	364				VTAM & NCP
		NN	366		G,L		APPN only
		OWNER	366		G,L		VTAM only; PUTYPE=1 or 2
		PDELAY	368		G,L	TERRESTR	APPN only
		PUDR	371		G,L		VTAM & NCP
		PUTYPE	371		G,L	5	VTAM & NCP
		RETRIES	373		G,L		VTAM & NCP
		SECNET	376		L	NO	VTAM
		SECURITY	376		G,L	UNSECURE	APPN only
		SEGMENT	377				VTAM & NCP

Table 31. NCP Major Node Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		TGN	381				VTAM & NCP
		TGP	382		G,L		APPN only
		TOPO	382			CONNECT	APPN only
		UPARM1	383		G,L	128	APPN only
		UPARM2	383		G,L	128	APPN only
		UPARM3	383		G,L	128	APPN only
		VERALSID	385		G,L	NO	APPN only
		XID	391		G,L	NO	VTAM & NCP
		XNETALS	391				XNETALS start option value
name	LU	LOCADDR	356	R			VTAM & NCP
		ASLENT	322		G,L,P		ASLTAB; VTAM
		ASLTAB	323		G,L,P		VTAM
		CERTIFY	330		G,L,P	NO	
		CKEYNAME	330			YES	VTAM
		CLRSESSQ	330		G,L,P	NO	VTAM
		DLOGMOD	337		G,L,P		VTAM
		ENCR	343		G,L,P	NONE	VTAM
		ENCRTYPE	343		G,L,P	DES	
		FASTPASS	344		G,L,P	YES	VTAM; TSO
		FEATUR2	345		G,L,P	(MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	VTAM
		ISTATUS	351		G	ACTIVE	VTAM
		LOGAPPL	356		G,L,P		VTAM
		LOGTAB	357		G,L,P		VTAM; dependent LUs
		LUDR	358		G,L,P		VTAM & NCP
		MAXSESS	360		G,L,P		VTAM & NCP Independent LUs only
		MDLENT	361		G,L,P		VTAM; MDLTAB
		MDLTAB	361		G,L,P		VTAM
		MODETAB	362		G,L,P	ISTINCLM	VTAM
		PACING	367		G,L,P	(1,1,ADAP)	VTAM & NCP
		REGISTER	372		P	see description	APPN only
		RESETLU	373		P	YES	
		RESSCB	373			0	VTAM & NCP
		SSCPFM	378		G,L,P	FSS	VTAM
		TERM	381		P		
		USSTAB	384		G,L,P	ISTINCDT	VTAM; dependent LUs
		VPACING	387		G,L,P	(2,1,ADAP)	VTAM
name	CLUSTER	GPOLL	347				VTAM & NCP
		ISTATUS	351		G	ACTIVE	VTAM
name	TERMINAL	TERM	381	R	G,L,C		VTAM & NCP
		ASLENT	322		G,L,C		ASLTAB

Table 31. NCP Major Node Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		ASLTAB	323		G,L,C		
		BHSET	327		G,L,C		VTAM & NCP
		DLOGMOD	337		G,L,C		VTAM; MODETAB
		FEATUR2	345		G,L,C	(MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	VTAM
		ISTATUS	351		G	ACTIVE	VTAM
		LOGAPPL	356		G,L,C		VTAM; NetView or TSO
		LOGTAB	357		G,L,C		dependent LUs
		MDLENT	361		G,L,C		MDLTAB
		MDLTAB	361		G,L,C		
		MODETAB	362		G,L,C	ISTINCLM	
		SSCPFM	378		G,L,C	FSS	
		USSTAB	384		G,L,C	ISTINCDT	dependent LUs

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

L Code this operand on the higher-level LINE definition statement to take advantage of the sift effect, or code it on this definition statement.

P Code this operand on the higher-level PU definition statement to take advantage of the sift effect, or code it on this definition statement.

C Code this operand on the higher-level CLUSTER definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

After changing a VTAM-only or VTAM-restricted operand in an NCP definition statement, a copy of the updated NCP definition statements must also be filed as a VTAM major node.

Although you do not code all the information in the NCP definition statements, you should consider the possible requirements of VTAM and NCP when you code the NCP definition statements.

NODETYPE Start Option

If you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

Dynamic Change of Operands

You can dynamically change the following NCP major node operands by editing the VTAMLST data set and then issuing the VARY ACT,UPDATE=ALL command. You can specify the new operand value on a higher-level definition statement for sifting, if applicable.

PU
CAPACITY
COMRATE
COSTBYTE
COSTIME
CPCP
DISCNT
DLCI
DYNADJCP
DYNLU
DYNWIND
HPRQLIM
INNPORT
INTFACE
IPQLIM
ISTATUS
LADDR
LIMRES
LUGROUP
LUSEED
MAXSESS
NN
PDELAY
SECNET
SECURITY
SNETMASK
T1TIMER
T2TIMER
TGP
TOPO
UPARM1
UPARM2
UPARM3
VERALSID
VPACING

LU
ASLENT
ASLTAB
CLRSESSQ
DLOGMOD
ENCR
ENCRTYPE
FASTPASS
FEATUR2
ISTATUS
LOGAPPL
LOGTAB
MDLENT
MDLTAB
MODETAB
REGISTER
SSCPFM
TERM
USSTAB
VPACING

For information on the following NCP operands, from the preceding list, refer to the *NCP, SSP, and EP Resource Definition Reference*.

- COMRATE
- DLCI. DLCI has a VTAM restriction. See “DLCI” on page 337 for details.
- DYNWIND
- HPRQLIM
- INTFACE
- IPQLIM
- LADDR
- SNETMASK
- T1TIMER
- T2TIMER

To dynamically change the operand, the resource to which it applies must be inactive. For instance, an operand on an LU definition statement which receives its value from a PU definition statement can be dynamically changed only when the LU is inactive; the PU can remain active.

You can dynamically change the operands associated with tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) at any time without inactivating resources.

VTAM cannot recognize dynamically changed ADDRs or LOCADDRs on the first activation of an NCP, even when VARY ACT,UPDATE=ALL is used. (Warm starts are considered initial activations.)

The resource resolution table (RRT) from the NCP generation does not contain ADDR or LOCADDR information, so VTAM cannot recognize that the ADDR and LOCADDR values have been changed on the first activation of the NCP. However, on changes to ADDR and LOCADDR after the first activation, VTAM can compare the value in the definition statements to the value previously coded and recognize the change.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and *OS/390 SecureWay Communications Server: SNA Operation* for more information on the VARY ACT,UPDATE=ALL command.

Note: For an NCP major node, you cannot add APPN capability or SDDLU support to a PU with the dynamic change function. If a PU has not been coded as APPN-capable, you cannot add the APPN operands to it. Likewise, if you have not coded LUGROUP (which allows SDDLU support) on a PU definition statement, you cannot add it with dynamic change.

Full Syntax

The full syntax for the network control program (NCP) major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

PCCU

```
►►──┬──────────┬──PCCU──┬─AUTODMP=NO──────────┬──┬─AUTOIPL=NO──────────┬──►
    └─name─────┘        │                     │  │                     │
                        └─AUTODMP=─┬─NO──┬────┘  └─AUTOIPL=─┬─NO──┬────┘
                                   └─YES─┘                   └─YES─┘

►──┬─AUTOSYN=YES──────────┬──┬─BACKUP=NO──────────┬──►
   │                      │  │                    │
   └─AUTOSYN=─┬─YES──┬────┘  └─BACKUP=─┬─NO──┬────┘
              └─NO───┘                 └─YES─┘
```

```
►─┬─CDUMPDS=value_coded_on_dumpds─┬─┬─CHANCON=COND──────────┬──────────────────►
  └─CDUMPDS=cps_dump_name─────────┘ └─CHANCON=─┬─COND──┬─┘
                                               └─UNCOND─┘

►─┬─CONFGDS=confgds_name──────────────────────────────┬─┬─CUADDR=device_address─┬──►
  └─CONFGPW=confgds_password,CONFGDS=confgds_name─────┘ └───────────────────────┘

►─┬─DELAY=0──────────────┬─┬─DUMPDS=dump_name─┬─┬─DUMPLD=NO──────────┬──────────►
  └─DELAY=timeout_value──┘ └──────────────────┘ └─DUMPLD=─┬─NO──┬─┘
                                                          └─YES─┘

►─┬─DUMPSTA=link_station_name─┬─┬─GWCTL=SHR──────────┬─┬─INITEST=NO──────────┬──►
  └───────────────────────────┘ └─GWCTL=─┬─SHR──┬─┘ └─INITEST=─┬─NO──┬─┘
                                         └─ONLY─┘                └─YES─┘

►─┬─LOADFROM=HOST──────────┬─┬─LOADSTA=link_station_name─┬──────────────────────►
  └─LOADFROM=─┬─HOST───┬─┘ └───────────────────────────┘
              └─EXTERNAL─┘

►─┬─MAXDATA=65535──────────┬─┬─MDUMPDS=value_coded_on_dumpds─┬──────────────────►
  └─MAXDATA=number_of_bytes─┘ └─MDUMPDS=moss_dump_name────────┘

►─┬─NETID=network_id─┬─┬─NEWPATH=─┬─name──────────────────┬─┬──────────────────►
  └──────────────────┘ │          ├─(name)────────────────┤ │
                       │          ├─(name1,name2)─────────┤ │
                       │          └─(name1,name2,name3)───┘ │

►─┬─OWNER=owner_name─┬─┬─RNAME=─┬─link_station_name──────┬─┬──────────────────►
  └──────────────────┘ │        └─(-link_station_name──)─┘ │
                       │                                   │
```

```
                                                   (2)
                                              ┌TGN=ANY──────────┐
                                                   (1)
 ┌SAVEMOD=NO──────┐                           ├TGN=1────────────┤
►┤                ├──┬──────────────────┬──────┤                ├──────────►
 └SAVEMOD=─┬─NO─┬─┘  └SUBAREA=suba_num─┘      │          ┌ANY──┐
           │    │                             └TGN=──┬──1──┬─┘
           └YES─┘                                    └tg_number┘

 ┌VFYC=YES────────┐   ┌VFYLM=NO────────┐
►┤                ├───┤                ├──────────────────────────────►◄
 └VFYC=─┬─YES──┬─┘   └VFYLM=─┬─NO─┬─┘
        ├─IGNORE─┤            └YES─┘
        └NO────┘
```

Notes:

1 TGN defaults to 1 for subarea.

2 TGN defaults to ANY for APPN.

BUILD

```
►►──┬──────────┬──BUILD──LOADLIB=ddname──NETID=network_id──SUBAREA=suba_addr──────────►
    └name──────┘

 ┌COSTAB=ISTSDCOS──────────────────┐
►┤                                 ├──┬──────────┬──────────────────────────────►
 └COSTAB=class-of-service_table_name┘  └MAXSUBA=n─┘

                                                   ┌SALIMIT=255──────┐
►──┬──────────────────────────┬──┬──────────────┬──┤                 ├──────────►
   └NEWNAME=ncp_load_module───┘  └PUNAME=pu_name─┘  └SALIMIT=─┬─255──┬─┘
                                                              ├─511──┤
                                                              ├─1023─┤
                                                              ├─2047─┤
                                                              ├─4095─┤
                                                              ├─8191─┤
                                                              ├─16383┤
                                                              ├─32767┤
                                                              └─65535┘
```

```
►─┬─VTAMTOPO=REPORT──────────┬──────────────────────────────────────►◄
  │                          │
  └─VTAMTOPO=─┬─REPORT───┬───┘
              ├─NOREPORT─┤
              ├─NOLLINES─┤
              ├─NOSWPUS──┤
              ├─INCLUDE──┤
              └─IGNORE───┘
```

HOST

```
►►─┬──────┬─HOST─INBFRS=number_of_buffers─MAXBFRU=number_of_buffers──────►
   └─name─┘

►─┬──────────────────────┬─┬─SUBAREA=1──────────┬──────────────────────►◄
  └─NETID=network_id─────┘ │                    │
                           └─SUBAREA=─┬─1──────┬─┘
                                      └─suba_addr─┘
```

NETWORK

```
►►─┬──────┬─NETWORK─┬──────────────────────────┬──────────────────────►
   └─name─┘         └─COPIES=number_of_networks─┘

►─┬─COSTAB=ISTSDCOS──────────────────────┬─┬──────────┬─┬──────────────┬─►
  │                                      │ └─MAXSUBA=n─┘ └─NETID=network_id─┘
  └─COSTAB=class-of-service_table_name───┘

►─┬─SALIMIT=255──────────┬─SUBAREA=suba_addr──────────────────────────►◄
  │                      │
  └─SALIMIT=─┬─255────┬──┘
             ├─511────┤
             ├─1023───┤
             ├─2047───┤
             ├─4095───┤
             ├─8191───┤
             ├─16383──┤
             ├─32767──┤
             └─65535──┘
```

FRSESET

```
►►──┬──────┬──FRSESET──SUBPORTS=──┬─(puname1,puname2)──────────────────┬──►◄
    └name──┘                      ├─(puname1,puname2,puname3,)─────────┤
                                  ├─(puname1,puname2,puname3,puname4)──┤
                                  ├─(puname1,puname2,,puname4)─────────┤
                                  └─(puname1,puname2,,)────────────────┘
```

GROUP

```
►►──name──GROUP──┬─DIAL=NO──────────┬──┬─DYNPU=NO──────────┬──┬─DYNPUPFX=CN──────────┬──►
                 └─DIAL=──┬─NO──┬─┘    └─DYNPU=──┬─NO──┬─┘    └─DYNPUPFX=string──┘
                          └─YES─┘                └─YES─┘

►──┬──────────┬──┬─ENCR=NONE────────┬──┬─ENCRTYPE=DES──────┬──►
   └ECLTYPE───┘  └─ENCR=──┬─NONE──┬─┘  └─ENCRTYPE=TDES24──┘
                          ├─OPT───┤
                          ├─REQD──┤
                          └─SEL───┘

►──┬─FRELAY=(PHYSICAL──┬──────────────┬──)──┬─ISTATUS=ACTIVE────────┬──►
   │                   ├─,PERIPHERAL──┤     └─ISTATUS=──┬─ACTIVE───┬─┘
   │                   ├─,SUBAREA─────┤                 └─INACTIVE──┘
   │                   └─,ANY─────────┘
   └─FRELAY=(LOGICAL──┬──────────────┬──)──┘
                      ├─,PERIPHERAL──┤
                      ├─,SUBAREA─────┤
                      └─,IP──────────┘

►─LNCTL=──┬─BSC──┬──┬──────────────────────────────────┬──┬─PUTYPE=5──────────┬──►
          ├─CA───┤  └─PHYSRSC=──┬─pu_name──────────────┬─┘  └─PUTYPE=──┬─5──┬─┘
          ├─SDLC─┤              ├─NONE─────────────────┤                ├─1──┤
          ├─SS───┤              └─(──NONE──────────)──┘                ├─2──┤
          └─USER─┘                  ├─,──TIC3──┤                       └─4──┘
                                    └─,──ODLC──┘

►──┬─SHM=NO──────────┬──┬──────────────────────────────────┬──►◄
   └─SHM=──┬─NO──┬─┘    └─VTAMTOPO=──┬─REPORT───────────┬─┘
           └─YES─┘                   ├─NOREPORT─────────┤
                                     ├─NOLLINES─────────┤
                                     └─NOSWPUS──────────┘
```

LINE

```
►►─name──LINE─┬─ANSWER=ON──────────┬──────────────────────────────────────────►
              │                    │
              └─ANSWER=─┬─ON──┬────┘
                        └─OFF─┘

►─┬─AUTODL=YES──────────┬─┬─BUFCAP=1──────────────┬─┬─CALL=IN──────────────┬──►
  │                     │ │                       │ │                      │
  └─AUTODL=─┬─YES─┬─────┘ └─BUFCAP=number_of_pius─┘ └─CALL=─┬─IN────┬────────┘
            └─NO──┘                                         ├─INOUT─┤
                                                            └─OUT───┘

►─┬─CKEYNAME=name──────────────┬─┬─ENCR=NONE──────────┬─┬─ENCRTYPE=DES──────┬──►
  │                            │ │                    │ │                    │
  └─CKEYNAME=crypto_key_name───┘ └─ENCR=─┬─NONE──┬────┘ └─ENCRTYPE=TDES24──┘
                                         ├─OPT───┤
                                         ├─REQD──┤
                                         └─SEL───┘

►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────────────────
  │                         │
  └─ISTATUS=─┬─ACTIVE──┬────┘
             └─INACTIVE─┘

──┬─────────────────┬──LOCADD=address──────────────────────────────────────────►
  └─LINEAUT=YES─────┘

►─┬─POLIMIT=(1,NOWAIT)──────────────────────────┬─┬─PU=YES──────────┬──────►
  │                                             │ │                 │
  └─POLIMIT=(─┬─1──┬──┬─,NOWAIT──────────────┬─)─┘ └─PU=─┬─YES─┬─────┘
              └─n──┘  │                      │           └─NO──┘
                      └─,─┬─NOWAIT─┬──────┘
                          ├─QUEUE──┤
                          └─WAIT───┘

►─┬──────────────────────────────────────────┬─┬─────────────────┬──────────►
  └─RETRIES=(,pause_interval,maximum_retries,)─┘ └─TERM=type──────┘

►─┬─USE=NCP──────────┬──────────────────────────────────────────────────────►
  │                  │
  └─USE=─┬─NCP───┬───┘
         ├─EP────┤
         ├─REDEF─┤
         └─SPARE─┘

►─┬──────────────────────────────────────────────────────────────────────────┬►
  └─VNNAME=connection_network_cpname,VNGROUP=group_name──────────────────────┘
```

NCP Major Node

```
►──┬──────────────────────────────────────────────────────────────────────────────────►◄
   │                 ┌─REPORT───┐
   └─VTAMTOPO=───┬───┴──────────┴──┬──────────────────────────────────────────────────
                 └─NOREPORT─────────┘
```

PU

```
                                                                   ┌─ANS=STOP──────────┐
►►─name──PU──┬──────────────────────────────┬──┬──────────────────┴───────────────────┴──►
             └─ADDR=link_station_address────┘  │                 ┌─STOP────┐
                                               └─ANS=────┬───────┴─────────┴─────────┬──
                                                         └─CONTINUE────────────────────┘

                                                   (1)
                                   ┌─CAPACITY=4M──────────┐
►──┬─AUTHLEN=YES──────┬──┬──────┬─┴─CAPACITY=8K──────────┴──────────────────────────────►
   │                  │  │      │
   └─AUTHLEN=──┬─NO─┬───┘  │      └─CAPACITY=number_of_bits──────────────────────────────
               └─YES┘      │
                           │
   ┌─CONNTYPE=value_of_conntype_start_option──────────────────┐
►──┼──────────────────────────────────────────────────────────┴──────────────────────────►
   │                 ┌─value_of_conntype_start_option──┐
   │  └─CONNTYPE=────┼─APPN────────────────────────────┼──┬──────────────────────────────
   │                 └─LEN─────────────────────────────┘  │
   │  ┌─CONNTYPE=APPN──┐  (2)                            │
   └──┴────────────────┴─────────────────────────────────┘
      └─CONNTYPE=APPN──┘

►──┬─COSTBYTE=0──────────────┬──┬─COSTTIME=0──────────────────┬──────────────────────────►
   │                         │  │                             │
   └─COSTBYTE=cost_per_byte──┘  └─COSTTIME=cost_per_unit_time──┘

   ┌─CPCP=value_of_cpcp_start_option──────┐
►──┼──────────────────────────────────────┴──┬──────────────────────────────────────────►
   │                 ┌─value_of_cpcp_start_option──┐  │  └─CPNAME=control_point_name──┘
   └─CPCP=───────────┼─NO──────────────────────────┼──┘
                     └─YES─────────────────────────┘

   ┌─DATMODE=HALF──────┐  ┌─DISCNT=(NO)──────────────────────┐
►──┼───────────────────┴──┼──────────────────────────────────┴──┬──────────────────────────►
   │                 ┌─HALF─┐  │                 ┌─(NO)──,F──┐  │  └─DLCI=address──┘
   └─DATMODE=────────┴──────┴──┘  └─DISCNT=────┬───┴───────────┴──┘
                     └─FULL─┘                  └─(YES──┬─────┬──)
                                                       ├─,NF─┤
                                                       └─────┘
```

```
  ┌─DYNADJCP=value_of_dynadjcp_start_option──────┐
  │                                              │
  │                ┌─value_of_dynadjcp_start_option─┐
  └─DYNADJCP=──────┼──────────────────────────────┼
                   ├─NO───────────────────────────┤
                   └─YES──────────────────────────┘
  ┌─DYNADJCP=YES─┐   (3)
  │              │
  └─DYNADJCP=YES─┘

  ┌─DYNLU=value_of_dynlu_start_option──────┐   ┌─ENCR=NONE──────┐
  │                                        │   │                │
  │            ┌─value_of_dynlu_start_option─┐   │            ┌─NONE─┐
  └─DYNLU=─────┼────────────────────────────┼   └─ENCR=──────┼──────┼
               ├─NO───────────────────────┤                   ├─OPT──┤
               └─YES──────────────────────┘                   ├─REQD─┤
  ┌─DYNLU=YES─┐   (4)                                         └─SEL──┘
  │           │
  └─DYNLU=YES─┘

  ┌─ENCRTYPE=DES────┐
  │                  │
  └─ENCRTYPE=TDES24──┘

                                                   (5)
  ┌─GP3174=NO──────────────┐   ┌─HPR=YES──────┐   ┌─INCLUD0E=NO──────┐
  │                        │   │              │   │                  │
  │            ┌─NO─────────┐   └─HPR=──┬─NO─┬─┘   └─INCLUD0E=──┬─NO─┬─┘
  └─GP3174=────┼────────────┼                   │               └─YES┘
               └─group_address┘                   └─YES┘

  ┌─INNPORT=NO──────┐   ┌─ISTATUS=ACTIVE──────┐   ┌─LIMRES=NO──────┐
  │                 │   │                     │   │                │
  │            ┌─NO─┐   └─ISTATUS=──┬─ACTIVE─┬─┘   │            ┌─NO─┐
  └─INNPORT=───┼────┼                │        │     └─LIMRES=────┼────┼
               └─YES┘                └─INACTIVE┘                  └─YES┘
                                                   ┌─LIMRES=NO─┐   (6)
                                                   │           │
                                                   └─LIMRES=NO─┘

  ┌─LLERP=NOTPREF────┐
  ├─LLERP=REQUIRED───┤   (7)
  │                  │
  └─LLERP=──┬─REQUIRED─┬─┘
            └─NOTPREF──┘

                                                   ┌─LPDA=ALLOW──────┐
  ┌──────────────────────────────────────────────┐ │                 │
  │                                              │ │            ┌─ALLOW─┐
  └─LMODADR=──┬─seg1_loc_mod_addr──────────────┬─┘ └─LPDA=──────┼───────┼
              └─(seg1_loc_mod_addr,seg2_loc_mod_addr)─┘            └─BLOCK─┘
```

NCP Major Node

```
►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──►
   └LUGROUP=lu_group_name┘  └LUSEED=lu_pattern_name┘  └NATIVE=──┬──NO──┬──┘
                                                                 └──YES─┘

►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──►
   └NETID=network_id──────┘  └NN=──┬──NO──┬──┘  └OWNER=──┬──owner_name──────┬──┘
                                   └──YES─┘              └──(──owner_name──)─┘

                ┌──PDELAY=TERRESTR──────┐
                │                (8)    │
                └──PDELAY=NEGLIGIB──────┘
►───────────────────────────────────────────────────────────────────────────────►
                ┌──TERRESTR──┐
                │            │
                ├──LONG──────┤
                │            │
                ├──NEGLIGIB──┤
                │            │
                └──PACKET────┘
                └──PDELAY=────┘

►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──►
   └PUDR=──┬──NO──┬──┘  └PUTYPE=5──────────┘  └PUTYPE=──┬──5──┬──┘
           └──YES─┘                                      ├──1──┤
                                                         ├──2──┤
                                                         └──4──┘

►──┬──────────────────────────────────────────────┬──┬──────────────────────┬──►
   └RETRIES=(,pause_interval,maximum_retries,)────┘  └SECNET=NO──────────┘
                                                     └SECNET=──┬──NO──┬──┘
                                                               └──YES─┘

                                                          (10)
                                                     ┌──TGN=ANY──────────┐
                                                     │        (9)        │
                ┌──SECURITY=UNSECURE──────┐          ├──TGN=1────────────┤
►───────────────────────────────────────────────────────────────────────────────►
                │                         │          │                   │
                └──SECURITY=──┬──UNSECURE──┬──┘  └SEGMENT=──┬──1──┬──┘  └TGN=──┬──ANY──┬──┘
                              ├──PUBLIC────┤                └──2──┘            ├──1────┤
                              ├──UNDERGRO──┤                                   │       │
                              ├──SECURE────┤                                   └──tg_number──┘
                              ├──GUARDED───┤
                              ├──ENCRYPT───┤
                              └──SHIELDED──┘
                                                     ┌──UPARM1=128──────────────────────┐
►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────────────┬──►
   └TGP=tgp_name──────────┘  └TOPO=CONNECT──────┘  └UPARM1=user-defined_operand_value┘

                ┌──UPARM2=128──────────────────────┐  ┌──UPARM3=128──────────────────────┐
►───────────────────────────────────────────────────────────────────────────────────────────►
                └──UPARM2=user-defined_operand_value┘  └──UPARM3=user-defined_operand_value┘
```

```
►─┬─┬─VERALSID=NO──────────┬─┬─XID=NO──────────┬──────────────────────────►
  │ │                      │ │                 │
  │ └─VERALSID=─┬─NO─┬─────┘ └─XID=─┬─NO─┬─────┘
  │             └─YES┘              └─YES┘
  │ ┌─VERALSID=NO─┐  (11)
  └─┤             │
    └─VERALSID=NO─┘

►─┬─XNETALS=value_of_XNETALS_start_option─┬──────────────────────────────►◄
  └─XNETALS=─┬─NO─┬──────────────────────┘
             └─YES┘
```

Notes:

1 For connections to virtual nodes, the default value is 4M.

2 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

3 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

4 DYNLU=YES is the only valid value if the TRLE operand is coded.

5 HPR defaults to NO when the HPR start option is set to NONE, (RTP,NONE), or (ANR,NONE).

6 LIMRES=NO is the only valid value if the TRLE operand is coded.

7 For type 2.1, the default is LLERP=REQUIRED. For SDLC, X.25, and channels links, the default is required. Otherwise, the default is LLERP=NOTPREF.

8 NEGLIGIB is the default if the TRLE operand is coded.

9 TGN defaults to 1 for subarea.

10 TGN defaults to ANY for APPN.

11 VERALSID=NO is the only valid value if the TRLE operand is coded.

LU

```
►►─name──LU──LOCADDR=address──┬──────────────────────────────────────────►
                              └─ASLENT=associated_lu_table_entry_name─┘

►─┬──────────────────────────────┬─┬─CERTIFY=NO──────────┬──────────────►
  └─ASLTAB=associated_lu_table_name─┘ │                     │
                                      └─CERTIFY=─┬─NO─┬─────┘
                                                 └─YES┘
```

NCP Major Node

```
►─┬─CKEYNAME=name──────────────────┬─┬─CLRSESSQ=NO──────────┬──────────────────────────►
  └─CKEYNAME=crypto_key_name───────┘ └─CLRSESSQ=─┬─NO──┬────┘
                                                 └─YES─┘

►─┬─DLOGMOD=default_logon_mode_table_entry_name──┬─┬─ENCR=NONE────────────┬──────────────►
                                                 └─ENCR=─┬─NONE──┬────────┘
                                                         ├─OPT───┤
                                                         ├─REQD──┤
                                                         └─SEL───┘

►─┬─ENCRTYPE=DES────────┬─┬─FASTPASS=YES─────────┬──────────────────────────────────────►
  └─ENCRTYPE=TDES24──────┘ └─FASTPASS=─┬─YES─┬────┘
                                       └─NO──┘

►─┬─FEATUR2=(MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────────────────────┬──►
  └─FEATUR2=(─┬─MODEL1──┬─┬─NOEDATS──┬─┬─NOPRINTR──┬─┬─NOSELPEN──┬─)──────────────────┘
              └─MODEL2──┘ └─EDATS────┘ └─PRINTR────┘ └─SELPEN────┘

►─┬─ISTATUS=ACTIVE───────────────┬─┬─LOGAPPL=controlling_primary_lu─┬──────────────────►
  └─ISTATUS=─┬─ACTIVE──┬──────────┘ └────────────────────────────────┘
             └─INACTIVE─┘

►─┬─LOGTAB=interpret_table_name──┬─┬─LUDR=YES─────────────┬──────────────────────────────►
                                 └─LUDR=─┬─YES─┬──────────┘
                                         └─NO──┘

►─┬─MAXSESS=max_number_LU-LU_sessions──┬──────────────────────────────────────────────►
  └────────────────────────────────────┘

►─┬─MDLENT=model_name_table_entry_name──┬─┬─MDLTAB=model_name_table_name──┬──────────────►
  └─────────────────────────────────────┘ └───────────────────────────────┘

►─┬─MODETAB=ISTINCLM──────────────────┬──────────────────────────────────────────────────►
  └─MODETAB=logon_mode_table_name─────┘
```

```
                                                        (2)
                                                 ┌REGISTER=NO──────────┐
                                                 │                (1)  │
 ┌PACING=(1,1,ADAP)──────────────────────────┐   │REGISTER=NETSRVR─────┤
►┤                                            ├───┤                     ├──►
 └PACING=┬(1,1,ADAP)──────────────────────┬┘   │              ┌NO────┐
         ├max──────────────────────────────┤    └REGISTER=─┬NETSRVR┤
         ├(max──────────────────────────)──┤               └CDSERVR┘
         │     ├,,ADAP──────────┤          │
         │     ├,,FIXED─────────┤          │
         │     └,min─┬ADAP┬─────┘          │
         │           └FIXED┘                │
         └(┬,,ADAP┬──────────)──────────────┘
           └,,FIXED┘
 ┌RESETLU=YES──┬,,ADAP┬──┬RESSCB=0──────────────────────────────┐
►┤             └,,FIXED┘  │                                      ├──►
 └RESETLU=┬YES┬──────────┘ └RESSCB=number_of_session_control_blocks┘
          └NO┘

 ┌SSCPFM=FSS──────────┐                  ┌USSTAB=ISTINCDT──────┐
►┤                     ├──┬TERM=type┬────┤                     ├──►
 └SSCPFM=┬FSS┬──────────┘  └────────┘    └USSTAB=uss_table_name┘
         ├USSNTO┤
         ├USSSCS┤
         ├USS3270┤
         ├USS3275┤
         └USS3780┘

 ┌VPACING=(2,1,ADAP)──────────────────────────┐
►┤                                             ├──────────────────────────►◄
 └VPACING=┬(2,1,ADAP)──────────────────────┬┘
          ├max──────────────────────────────┤
          ├(max──────────────────────────)──┤
          │     ├,,ADAP──────────┤          │
          │     ├,,FIXED─────────┤          │
          │     └,min─┬ADAP┬─────┘          │
          │           └FIXED┘                │
          └(┬,,ADAP┬──────────)──────────────┘
            └,,FIXED┘
```

Notes:

1 REGISTER defaults to NETSRVR for dependent LUs.

2 REGISTER defaults to NO for independent LUs.

CLUSTER

```
►►──name──CLUSTER──┬──────────────────────────┬──────────────────────────────►◄
                   └─GPOLL=polling_character─┘
```

TERMINAL

```
►►──name──TERMINAL──TERM=type──┬──────────────────────────────────────────┬──►
                               └─ASLENT=associated_lu_table_entry_name─┘

►──┬──────────────────────────────┬──┬─BHSET=NONE──────────────┬──────────►
   └─ASLTAB=associated_lu_table_name─┘  │                         │
                                        └─BHSET=─┬─NONE──────┬─┘
                                                 ├─DYNAMIC───┤
                                                 └─setname───┘

►──┬──────────────────────────────────────────────────────────┬──────────────►
   └─DLOGMOD=default_logon_mode_table_entry_name─┘

   ┌─FEATUR2=(MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────┐
►──┼──────────────────────────────────────────────────────────────────────┼──►
   └─FEATUR2=(─┬─MODEL1──┬──┬─NOEDATS─┬──┬─NOPRINTR─┬──┬─NOSELPEN─┬─)──┘
               └─MODEL2──┘  └─EDATS───┘  └─PRINTR───┘  └─SELPEN───┘

   ┌─ISTATUS=ACTIVE──────────┐
►──┼─────────────────────────┼──┬──────────────────────────────────────────►
   └─ISTATUS=─┬─ACTIVE──┬──┘  └─LOGAPPL=controlling_primary_lu─┘
              └─INACTIVE─┘

►──┬──────────────────────────────┬──┬──────────────────────────────────────►
   └─LOGTAB=interpret_table_name─┘  └─MDLENT=model_name_table_entry_name─┘

►──┬──────────────────────────────┬──┬─MODETAB=ISTINCLM──────────────┬──────►
   └─MDLTAB=model_name_table_name─┘  │                               │
                                     └─MODETAB=logon_mode_table_name─┘

   ┌─SSCPFM=FSS──────────┐  ┌─USSTAB=ISTINCDT──────────┐
►──┼─────────────────────┼──┼──────────────────────────┼──────────────────►◄
   └─SSCPFM=─┬─FSS─────┬─┘  └─USSTAB=uss_table_name──┘
             ├─USSNTO──┤
             ├─USSSCS──┤
             ├─USS3270─┤
             ├─USS3275─┤
             └─USS3780─┘
```

Operand Descriptions

The following section describes the operands you can code for the network control program (NCP) major node.

ADDR

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─ADDR=link_station_address─┘
```

statements: *PU*
dependencies: *VTAM and NCP; DLCI*

Specifies the 2-character hexadecimal representation of the 8-bit address of a PU on a nonswitched line. Any bit configuration except X'00' or X'FF' is valid.

ADDR cannot be coded on the first PU under a frame relay line, but needs to be coded on the other PUs.

Note: VTAM cannot recognize a dynamically changed ADDR on the first activation of an NCP, even when VARY ACT,UPDATE=ALL is used. (Warm starts are considered initial activations.)

The resource resolution table (RRT) from the NCP generation does not contain ADDR information, so VTAM cannot recognize that the ADDR value has been changed on the first activation of the NCP. However, on changes to ADDR after the first activation, VTAM can compare the value in the definition statement to the value previously coded and recognize the change.

Note: You can code ADDR or DLCI on frame relay PUs, but not both.

ANS

```
    ┌─ANS=STOP──────────┐
►►──┼──────────────────┼──────────────────────────────────────────────────►◄
    │        ┌─STOP────┐│
    └─ANS=───┼─────────┼┘
             └─CONTINUE┘
```

statements: *GROUP, LINE, PU*
dependencies: *VTAM and NCP*

Determines whether the physical unit continues to operate when the NCP enters automatic network shutdown.

ANS=CONTINUE or CONT
Specifies the LU-LU session is to continue when the NCP enters into automatic network shutdown mode. Code ANS=CONTINUE

- If the terminal participates in XRF sessions
- For frame relay PUs

When a session is continued and takeover occurs, if the LU taken over is a dependent logical unit, it must support ACTLU(ERP), or the session is terminated during takeover processing. An independent logical unit does not

receive ACTLU, so this restriction does not apply. Note that there are several restrictions for session continuity in certain configurations. For information on these restrictions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

ANS=STOP
Specifies that the LU-LU session will be terminated when the NCP enters automatic network shutdown mode.

Refer to the *NCP, SSP, and EP Resource Definition Reference* to code ANS for BSC and SS resources.

ANSWER

```
►►─┬─ANSWER=ON──────────┬──────────────────────────────────────────►◄
   │                    │
   └─ANSWER=─┬─ON──┬────┘
             └─OFF─┘
```

statements: *GROUP, LINE*
dependencies: *VTAM only*

Specifies whether physical units can dial in to the NCP. ANSWER is valid only if the SDLC line has dial-in capability.

ANSWER=OFF
The physical units cannot dial in to the NCP, regardless of whether the line is active or inactive.

ANSWER=ON
The physical unit can dial in to the NCP when the line is activated.

The operator can issue a VARY ANS command after activation to change the answer state of the line. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ANS command.

ASLENT

```
►►─┬──────────────────────────────────────────┬──────────────────────►◄
   └─ASLENT=associated_lu_table_entry_name────┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only; ASLTAB*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─ASLTAB=associated_lu_table_name─┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or VTAM can send them to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

AUTO

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─AUTO=auto_call_unit_address─┘
```

statements: *LINE*
dependencies: *VTAM and NCP*

Specifies the address of an automatic calling unit for this line.

When AUTO is coded, VTAM sends the dial digits to NCP. Otherwise, VTAM issues a console message to inform the operator that a manual dial should be done.

AUTHLEN

```
    ┌─AUTHLEN=YES──────┐
►►──┤                  ├──────────────────────────────────────────────►◄
    └─AUTHLEN=──┬─NO──┬─┘
                └─YES─┘
```

statements: *GROUP, LINE, PU*
dependencies: *Type 2.1*

Specifies whether VTAM should pass the transmission priority field specified by this PU to another PU.

AUTHLEN=NO
VTAM will not pass the specified transmission priority, instead the default priority of medium will be passed.

AUTHLEN=YES
VTAM will pass the specified transmission priority.

AUTODL

```
►►─┬─AUTODL=YES──────────┬──────────────────────────────────────────►◄
   │                     │
   └─AUTODL=─┬─YES─┬─────┘
             └─NO──┘
```

statements: *GROUP, LINE*
dependencies: *VTAM and NCP*

Specifies whether CCITT X.21 data circuit-terminating equipment (DCE) accepts dial digits.

AUTODL=NO
Limits an X.21 line to direct-call-only status. As a result, when VTAM searches a line group for an available line, it cannot select a direct-call-only line for an auto-call operation.

AUTODL=YES
Indicates that the NCP is to run the normal callout procedure for the line.

Note: Code AUTODL=YES for switched subarea connections using X.21 lines.

AUTODMP

```
►►─┬─AUTODMP=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─AUTODMP=─┬─NO──┬─────┘
              └─YES─┘
```

statements: *PCCU*
dependencies: *valid only when DUMPDS is coded*

Determines whether a dump of communication controller storage is taken automatically after an unrecoverable failure in the NCP or the communication controller.

AUTODMP=NO
Indicates that the VTAM operator is to be asked whether a dump should be taken.

AUTODMP=YES
Indicates the dump is taken automatically.

The AUTODMP operand is valid only if you code a DUMPDS operand in this PCCU definition statement. Otherwise, no NCP dump can be taken. (See “DUMPDS” on page 338 for a description of DUMPDS.) After the dump, (or dumps, if more than

one host takes a dump), VTAM interprets the AUTOIPL setting to determine whether it should load another copy of the same NCP and restart it.

If the operator specifies DUMPLOAD=YES in the VARY ACT command for a 3720 or 3745 communication controller, or if DUMPLD=YES is specified on this PCCU definition statement, the communication controller attempts to dump and load the NCP. If the dump and load by the communication controller is successful, the AUTODMP and AUTOIPL operands are ignored.

In an NCP for a communication controller that is attached by multiple channels, more than one PCCU definition statement can have AUTODMP=YES.

Note: All VTAMs in the network must be at V3R4.1 or higher for this function to be used. Any mixture of back-level VTAMs including V3R4 VM, V3R4 VSE, or V3R4 MVS/ESA™ make this function not valid.

AUTOIPL

```
►►─┬─AUTOIPL=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─AUTOIPL=─┬─NO──┬────┘
              └─YES─┘
```

statements:	*PCCU*
dependencies:	*none*

Determines whether VTAM loads another copy of the NCP and restarts it after either an unrecoverable failure occurs in the NCP or the communication controller.

AUTOIPL=NO
Indicates that the VTAM operator is prompted on whether VTAM should load and restart another copy of the same NCP.

AUTOIPL=YES
Indicates that another copy of the same NCP is automatically loaded and restarted.

In an NCP for a communication controller that is attached by multiple channels, more than one PCCU definition statement can have AUTOIPL=YES.

If the operator specifies DUMPLOAD=YES in the VARY ACT command for a 3720 or 3745 communication controller, or DUMPLD=YES on this statement, the communication controller attempts to dump and load the NCP. If the dump and load by the communication controller is successful, the AUTODMP and AUTOIPL operands are ignored.

Note: All VTAMs in the network must be at V3R4.1 or higher for the IPL/Dump Contention function to be used. Any mixture of back-level VTAMs including V3R4 VM, V3R4 VSE, or V3R4 MVS/ESA make this function not valid.

AUTOSYN

```
┌─AUTOSYN=YES──────┐
►►──┼──────────────┼──────────────────────────────────────────►◄
└─AUTOSYN=─┌─YES─┐─┘
           └─NO──┘
```

statements: *PCCU*
dependencies: *none*

Used to determine whether the NCP load module specified replaces the current load module in the communication controller. This operand applies only if the VTAM operator requests (or uses by default) LOAD=U on the VARY ACT command for an NCP. VTAM uses this operand when the specified load module name matches the current load module name (defined by NEWNAME on the BUILD definition statement).

AUTOSYN=NO
The operator is asked whether the communication controller is to be refreshed with the specified load module.

AUTOSYN=YES
The communication controller will not be loaded with the specified load module.

When NEWNAME does not match VTAM's major node name, the value of VFYLM determines whether the controller is automatically reloaded. (See "VFYLM" on page 386 for a description of VFYLM.)

BACKUP

```
┌─BACKUP=NO────────┐
►►──┼──────────────┼──────────────────────────────────────────►◄
└─BACKUP=─┌─NO──┐─┘
          └─YES─┘
```

statements: *PCCU*
dependencies: *OWNER*

Determines whether this NCP's resources can be taken over by a backup host if the owning host fails.

BACKUP=NO
Indicates that the backup host cannot take over the NCP's resources.

BACKUP=YES
Indicates that a backup host can take over the NCP's resources.

If BACKUP=YES, resources are held in reserve unless they are owned by the host VTAM to which this NCP is defined. (Such resources have the same OWNER name as the one on this PCCU definition statement.) The host VTAM cannot use the resources until a VTAM VARY ACQ command is issued for this NCP.

Note: If the OWNER operand is not coded, BACKUP is ignored. (See "OWNER" on page 366 for a description of OWNER.)

BHSET

```
►►─┬─BHSET=NONE──────────────┬──────────────────────────────────────────►◄
   │                         │
   └─BHSET=─┬─NONE────┬──────┘
            ├─DYNAMIC─┤
            └─setname─┘
```

statements: *GROUP, LINE, CLUSTER, TERMINAL*
dependencies: *VTAM and NCP*

Block handler sets cannot be dynamically selected for BSC 3270s defined to VTAM. The BHSET operand can be coded by other owners of the terminal provided that the dynamic BHSET selection option (BHSASSC) is selected on the SYSCNTRL definition statement.

BUFCAP

```
►►─┬─BUFCAP=1──────────────────┬──────────────────────────────────────────►◄
   │                           │
   └─BUFCAP=number_of_pius─────┘
```

statements: *GROUP, LINE*
dependencies: *VTAM only*
range: *1–255*

Specifies the maximum number of PIUs that can be read at one time from a line without retransmissions, assuming each PIU is the maximum length allowed (MAXBFRU(MAX)×IOBUFSIZE). If an individual PIU length is smaller than the maximum size allowed for a PIU, more PIUs than BUFCAP's value could be read in. The greatest number of buffers that VTAM will allocate is BUFCAP×MAXBFRU(MAX).

A retry value of 0 will not allow a type 2.0 or type 2.1 connection to become active. One retry is required to build the XID that will be exchanged.

Note: No matter what value you give for BUFCAP, VTAM uses only as many buffers as necessary to read in all incoming data.

For example, suppose that after initial startup, BUFCAP=35, IOBUFSIZE=100, MAXBFRU=(2,8), and all the PIUs are maximum size (each PIU would be 800 bytes long, requiring eight buffers). The NCP sends only MAXOUT PIUs at one time. If MAXOUT=7, the number of buffers required to have no retransmissions is 56. The BUFCAP operand allows up to 280 (35 × 8), but they will not be used.

CALL

```
   ┌─CALL=IN──────────┐
►►─┼──────────────────┼──────────────────────────────────────────►◄
   │          ┌─IN────┐│
   └─CALL=────┼───────┼┘
              ├─INOUT─┤
              └─OUT───┘
```

statements:	*GROUP, LINE*
dependencies:	*VTAM and NCP*

Specifies whether VTAM, the device, or both can set up switched connections over the line to which this operand applies. The CALL operand acts as a security device.

CALL=IN
Specifies that the line is used only for terminal-initiated connections.

CALL=INOUT
Specifies that the line is used for connections initiated by terminals, in addition to connections initiated by VTAM, the VTAM operator, or application programs.

CALL=OUT
Specifies that the line is used only for outgoing calls.

Notes:

1. If you code CALL=OUT, or if you code CALL=INOUT without coding an AUTO operand, outgoing calls are dialed by the operator in response to a VTAM console message. For switched subarea lines with CALL=OUT or CALL=INOUT, AUTO is required. There are no VTAM restrictions on CALL=OUT.
2. This operand is valid only if DIAL=YES on the GROUP definition statement, and applies only to line operation in network control mode.
3. CALL=OUT is not supported by non-SNA terminals using Network Terminal Option (NTO).

CAPACITY

```
                 (1)
   ┌─CAPACITY=4M──────────────┐
   ├─CAPACITY=8K──────────────┤
►►─┼────────────────────────────┼──────────────────────────────────────────►◄
   └─CAPACITY=number_of_bits────┘
```

Notes:

1 For connections to virtual nodes, the default value is 4M.

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*
range:	*1K–1000M*

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either kilobits per second (for example, 100K) or megabits per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Note: Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CDUMPDS

```
   ┌─CDUMPDS=value_coded_on_dumpds─┐
►►─┤                               ├──────────────────────────────────────────►◄
   └─CDUMPDS=cps_dump_name─────────┘
```

statements: *PCCU*
dependencies: *none*

Names the file that contains communication scanner processor (CSP) dumps.

cps_dump_name must be on a DD statement, which defines the file that is to contain the data from a storage dump of a communication controller.

If CDUMPDS is not coded, the CSP dumps are contained in the file named on the DUMPDS operand.

Note: CDUMPDS applies only to 3720, 3725, and 3745 communication controllers.

CHANCON

```
   ┌─CHANCON=COND──────────┐
►►─┤                       ├──────────────────────────────────────────────►◄
   │              ┌─COND─┐ │
   └─CHANCON=─────┤      ├─┘
                  └─UNCOND─┘
```

statements: *PCCU*
dependencies: *none*

Determines whether VTAM sends a conditional or an unconditional channel contact request to the NCP when the host is to activate the NCP over a channel.

CHANCON=COND
VTAM allows the NCP to reject the host's contact request, if the host's subarea number is already in use by another host (attached to another one of the NCP's channels). This results if duplicate subareas are assigned to more than one host (attached to more than one channel interface). Two hosts might have inadvertently been assigned the same number, or they might have been assigned duplicate numbers to allow mutual backup capabilities.

CHANCON=UNCOND
The contact request is unconditional; the NCP must accept it. If another host is using the same subarea number, the NCP breaks contact with that host.

Note: If you plan to define the channel to the NCP in a channel-attachment major node, this operand is replaced by the CHANCON operand on the PU

definition statement for the channel-attachment major node. See “CHANCON” on page 72 for a description of the CHANCON operand in a channel-attachment major node.

CERTIFY

```
►►─┬─CERTIFY=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─CERTIFY=─┬─NO──┬────┘
              └─YES─┘
```

statements: *GROUP, LINE, PU, LU*
dependencies: *CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner's LU support also supports the CERTIFY function.*

Specifies whether VTAM on behalf of the application program will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CKEYNAME

```
►►─┬─CKEYNAME=name──────────────┬──────────────────────────────────►◄
   │                            │
   └─CKEYNAME=crypto_key_name───┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of the key-encrypting key (KEK) in the cryptographic key data set (CKDS) for the LU. Use CKEYNAME to encipher the session key that is sent in the BIND to the LU when a cryptographic session is established. CKEYNAME has the same format and restrictions as other VTAM *names*. For information on naming conventions, see “Format of Definition Statements” on page 9.

If you do not specify CKEYNAME, the cryptographic key name defaults to the name on the LU definition statement.

Note: Independent LUs will ignore the CKEYNAME keyword and use their default name.

CLRSESSQ

```
   ┌─CLRSESSQ=NO──────┐
►►─┤                  ├──────────────────────────────────────────►◄
   │              ┌─NO─┐
   └─CLRSESSQ=────┤    ├──┘
                  └─YES┘
```

statements: *GROUP, LINE, PU, LU*
dependencies: *VTAM only*

Specifies whether a dependent LU is forced into an inhibited session state when the LU is disabled.

When the LU is enabled, it is returned to normal status. The sessions, however, are not reconnected.

CLRSESSQ=NO
Specifies that the LU is not forced into inhibited session state.

CLRSESSQ=YES
Specifies that the LU is forced into an inhibited session state. Both active and pending sessions are removed from the session queue.

CONFGDS

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └─CONFGDS=name─┘
```

statements: *PCCU*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └─CONFGPW=password─┘
```

statements: *PCCU*
dependencies: *CONFGDS*

Specifies a 1–7 alphanumeric character password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

CONNTYPE

```
      ┌─CONNTYPE=value_of_conntype_start_option──────┐
►►────┼──────────────────────────────────────────────┼──────────────────────────►◄
      │              ┌─value_of_conntype_start_option─┐
      └─CONNTYPE=────┼────────────────────────────────┼
                     ├─APPN───────────────────────────┤
                     └─LEN────────────────────────────┘
      ┌─CONNTYPE=APPN─┐  (1)
      │               │
      └─CONNTYPE=APPN─┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*

Specifies for a type 2.1 whether the connection is to be established as a LEN connection or attempted as an APPN connection.

If you do not code the NODETYPE start option, the CONNTYPE operand is ignored, and the connection will be a LEN connection.

CONNTYPE=APPN
This connection can support parallel TGs, CP-CP sessions, and CP name change support. When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

CONNTYPE=LEN
Specifies that this connection does not support CP-CP sessions and CP name change, and that VTAM does not check for duplicate CPNAMEs.

Although you need to code CONNTYPE=APPN for a PU to be an APPN resource, this does not necessarily mean the PU will be APPN; a PU's characteristics are determined at activation. However, coding CONNTYPE=LEN means that the PU can only be a LEN resource, even if, at activation, the XID indicates the PU can be an APPN resource.

Because CONNTYPE=LEN indicates that the PU is not APPN-capable, do not code any APPN-only operands on the same PU statement. If an APPN-only operand is coded with CONNTYPE=LEN, VTAM will issue a message warning of a parameter conflict.

COPIES

```
►►──┬──────────────────────────┬──────────────────────────────────────────────►◄
    └─COPIES=number_of_networks─┘
```

statements:	*NETWORK*
dependencies:	*SNI only; NETID*

Specifies the number of unique networks that this network definition statement can support.

Note: NETID and COPIES cannot be coded on the same NETWORK definition statement.

COSTAB

```
►►─┬─COSTAB=ISTSDCOS──────────────────────────┬──────────────────────────────►◄
   │                                          │
   └─COSTAB=class-of-service_table_name───────┘
```

statements:	*BUILD, NETWORK*
dependencies:	*none*

Names a class-of-service (COS) table for the network identified by NETID. If you omit this operand, the default COS table is ISTSDCOS. This operand is processed regardless of which network corresponds to the NETWORK definition statement.

The COS table name matches the name of a class-of-service table in the VTAM load module library. VTAM loads the table from the library, associates it with the network identified by the NETID operand, and uses it to resolve class-of-service names for cross-network sessions when the primary logical unit is represented by an alias address in the gateway NCP.

This table is also used to resolve a class-of-service name to a virtual route (VR) list when the operator issues the command:

```
DISPLAY ROUTE,ORIGIN=ncp subarea node,COSNAME=cos name
```

COSTBYTE

```
►►─┬─COSTBYTE=0──────────────────┬──────────────────────────────────────────►◄
   │                             │
   └─COSTBYTE=cost_per_byte──────┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a cost-per-byte-transmitted to be associated with the transmission group (TG). A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
►►─┬─COSTTIME=0──────────────────────┬──────────────────────────────────────►◄
   │                                 │
   └─COSTTIME=cost_per_unit_time─────┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a cost-per-unit-of-time to be associated with the transmission group (TG). A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

CPCP

```
   ┌─CPCP=value_of_cpcp_start_option──────┐
►►─┤                                      ├──────────────────────────────────────►◄
   │          ┌─value_of_cpcp_start_option─┐
   └─CPCP=────┤                            ├──┘
              ├─NO───────────────────────────┤
              └─YES──────────────────────────┘
```

statements: *GROUP, LINE, PU*
dependencies: *APPN only*

Specifies whether CP-CP sessions are supported on this connection. If you do not code this operand, the value defaults to the value specified on the CPCP start option.

CPCP=NO
Indicates CP-CP sessions are not supported on this connection.

CPCP=YES
Indicates CP-CP sessions are supported on this connection.

CPNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────────►◄
    └─CPNAME=control_point_name─┘
```

statements: *PU*
dependencies: *none*

Identifies the resource's owning control point name. To allow definition of casually connected resources, this operand can be fully qualified with a network ID.

CUADDR

```
►►──┬──────────────────────┬──────────────────────────────────────────────────►◄
    └─CUADDR=device_address─┘
```

statements: *PCCU*
dependencies: *none*

Defines the 3-digit or 4-digit hexadecimal device number of the channel attachment for the communication controller in which the NCP runs.

If you omit CUADDR and the VTAM operator wants to activate the communication controller over a channel, the operator should provide an address in the U operand when issuing the VTAM VARY ACT command.

If no channel device address or link station contact points are coded (with the CUADDR and RNAME operands of the PCCU definition statement) and none are provided on the VARY ACT command, a link station adjacent to the communication controller must already be active and defined to the host before the operator can activate the controller.

Note: If you plan to define the channel to the NCP in a channel-attachment major node, this operand is replaced by the ADDRESS operand on the LINE definition statement for the channel-attachment major node. See “ADDRESS” on page 72 for a description of the ADDRESS operand in a channel-attachment major node.

DATMODE

```
►►─┬─DATMODE=HALF──────────┬──────────────────────────────────────────►◄
   │                       │
   └─DATMODE=─┬─HALF─┬─────┘
              └─FULL─┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*VTAM and NCP*

Specifies to the NCP whether the physical unit communicates in half-duplex or full-duplex data mode. The associated line must support simultaneous transmission in both directions.

DATMODE=FULL
Specifies that the physical unit communicates in full-duplex mode.

DATMODE=HALF
Specifies that the physical unit communicates in half-duplex mode.

DELAY

```
►►─┬─DELAY=0──────────────┬──────────────────────────────────────────►◄
   │                      │
   └─DELAY=timeout_value─┘
```

statements:	*PCCU*
dependencies:	*none*
range:	*0-9.999*

Specifies the maximum amount of time, in seconds and thousandths of seconds, that VTAM should wait after getting low priority (that is, PIU transmission priority 0 or 1) data to transmit.

For more efficient operation, code this value as high as possible. For maximum speed across the interface, set this value to 0. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information.

Note: If you plan to define the channel to the NCP in a channel-attachment major node, this operand is replaced by the DELAY operand on the PU definition

statement for the channel-attachment major node. See “DELAY” on page 73 for a description of the DELAY operand in a channel-attachment major node.

DIAL

```
►►─┬─DIAL=NO──────────┬──────────────────────────────────────────►◄
   │                  │
   └─DIAL=─┬─NO──┬────┘
           └─YES─┘
```

statements: *GROUP*
dependencies: *VTAM and NCP*

Specifies whether the lines in the group require a dialed (switched) or leased connection.

DIAL=NO
Specifies that the lines require a leased (nonswitched) connection.

DIAL=YES
Specifies that the lines require a dialed connection. When DIAL=YES, the only operands you can code in the PU definition statements subordinate to the GROUP definition statement are PUTYPE, SPAN, and XID.

DISCNT

```
►►─┬─DISCNT=(NO)──────────────────┬──────────────────────────────►◄
   │                              │
   └─DISCNT=─┬─(NO)──,F─────────┬─┘
             └─(YES──────────)──┘
                  ├─,NF─┤
```

statements: *GROUP, LINE, PU, CLUSTER*
dependencies: *VTAM only*

Specifies when VTAM should end its SSCP-LU and SSCP-PU sessions and, when DISCNT=YES, whether to indicate final-use status in the DACTPU request unit when it deactivates a physical unit.

DISCNT=(NO)
If you code NO or use the default, VTAM terminates the SSCP-LU and SSCP-PU sessions when certain conditions are met. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of those conditions.

DISCNT=(YES)
If you code YES, VTAM automatically ends the SSCP-LU and SSCP-PU sessions when the last LU-LU session is ended.

DISCNT=(YES,F)

DISCNT=(YES,NF)
specifies whether VTAM indicates a final-use status in the DACTPU request unit

when it deactivates a physical unit as a result of DISCNT=YES. This value does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

If you code F or use the default, final-use status is indicated and the connection can be ended. If you code NF, not-final-use status is indicated and the connection should not be ended.

Each device has its own requirements regarding final-use status. To determine whether to code F or NF for a given device, consult the appropriate installation publication for the device.

DLCI

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─DLCI=address─────┘
```

statements: *PU*
dependencies: *VTAM and NCP; ADDR*

DLCI specifies the DLCI address of the frame-relay subport or internet host. All DLCI values associated with a single frame-relay physical line or a NTRI frame handler logical line must be unique. For example, if you use DLCI X'30' on a frame-relay physical line to identify a subarea terminating equipment subport (TESP) or a frame handler subport (FHSP), address X'30' cannot be used by a peripheral device to establish an incoming logical peripheral connection to NCP over that same physical line.

You should avoid using DLCI X'20' (decimal 32) when you define frame-relay peripheral links, because peripheral devices often use DLCI X'20' for incoming calls. When a subarea link TESP or FHSP uses this DLCI, the peripheral devices must be reconfigured to use a different DLCI.

Note: You can code DLCI or ADDR on frame relay PUs, but not both.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information on the DLCI operand.

DLOGMOD

```
►►──┬──────────────────────────────────────────┬──────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name─┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only; MODETAB*

Specifies the name of the logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on the logon mode entries, see “Logon Mode Table” on page 698.

DUMPDS

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─DUMPDS=dump_name─┘
```

statements:	*PCCU*
dependencies:	*none*

Specifies one of three operands (the others are CDUMPDS and MDUMPDS) that name dump files for this communication controller. DUMPDS applies to any communication controller. CDUMPDS and MDUMPDS apply only to 3720, 3725, and 3745 communication controllers.

- DUMPDS names the file that contains a storage dump of a communication controller. For a 3720, 3725, or 3745, this file also contains communication scanner processor (CSP) dumps and maintenance and operator subsystem (MOSS) dumps if you have not set up another file for that purpose.
- For a 3720, 3725, or 3745, CDUMPDS names the file that contains CSP dumps.
- For a 3720, 3725, or 3745, MDUMPDS names the file that contains MOSS dumps.

dump_name must be on a DD statement, which defines the file that is to contain the data from a storage dump of a communication controller.

Code DUMPDS if any of the following are true:

- AUTODMP=YES on this PCCU definition statement.
- AUTODMP=NO on this PCCU definition statement and the VTAM operator answers YES to the resulting prompt for a dump.
- The VTAM operator issues the MODIFY DUMP command without the DUMPDS operand.

To format and print the data in a dump file, use the NCP independent dump utility program described in the *NCP, SSP, and EP Diagnosis Guide*. To display an NCP dump that is stored on a 3720 or 3745 communication controller, use the DISPLAY NCPSTOR operator command described in *OS/390 SecureWay Communications Server: SNA Operation*.

DUMPLD

```
    ┌─DUMPLD=NO──────┐
►►──┼────────────────┼──────────────────────────────────────────►◄
    │          ┌─NO─┐│
    └─DUMPLD=──┼────┼┘
               └─YES┘
```

statements:	*PCCU*
dependencies:	*see notes*

Indicates whether an NCP that has abended should be automatically dumped to the 3720 or 3745 communication controller disk, and then reloaded from that disk.

DUMPLD=NO
Specifies that the NCP that has abended is not automatically dumped and then reloaded from that disk.

DUMPLD=YES
Specifies that the NCP that has abended is automatically dumped and then reloaded from that disk.

Notes:

1. The value of DUMPLD is used only if you do not specify DUMPLOAD on the VARY ACT command, and only if a load is actually performed.
2. If you code LOADFROM=EXTERNAL and SAVEMOD=NO on the PCCU definition statement, the initial load will not use the SAVEMOD value.
3. If you code LOADFROM=HOST and SAVEMOD=NO on the VARY ACT command, and code DUMPLD=YES on the PCCU definition statement, the value of DUMPLD is not used for the initial load.
4. If you specify DUMPLD=YES and allow LOADFROM to default to HOST and SAVEMOD to default to NO, DUMPLD is forced to NO unless you specify LOADFROM=EXT on the VARY ACT command.

Refer to *OS/390 SecureWay Communications Server: SNA Operation* for additional information on the VARY ACT command.

DUMPSTA

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └DUMPSTA=link_station_name─────┘
```

statements:	*PCCU*
dependencies:	*none*

Specifies an adjacent link station, channel- or link-attached, that is used to perform dump operations. If you do not use DUMPSTA on either the PCCU definition statement or the VTAM VARY ACT command, VTAM chooses a default link station whenever a dump is necessary. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how VTAM chooses a default link station.

For a channel-attached communication controller, if the link station name is not defined (in the channel attached major node), VTAM assigns a link station name by taking the device number and adding “-S”. If the device number is 3-digits, a leading zero is added to the name. For example, the device number 0C2 has the link station name of 00C2-S.

You can designate the same link station for both dumping and loading.

For more information on link stations and use of the VTAM VARY ACT command, refer to *OS/390 SecureWay Communications Server: SNA Operation*.

DYNADJCP

```
      ┌─DYNADJCP=value_of_dynadjcp_start_option──────┐
►►────┤                                              ├──────────────────────────────►◄
      │              ┌─value_of_dynadjcp_start_option─┐
      └─DYNADJCP=────┤                                ├──┘
                     ├─NO──────────────────────────────┤
                     └─YES─────────────────────────────┘
      ┌─DYNADJCP=YES─┐   (1)
      │              ├──────────────────────────────────┘
      └─DYNADJCP=YES─┘
```

Notes:

1 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

statements: *GROUP, LINE, PU*
dependencies: *APPN only; ignored if CPNAME coded.*

Specifies whether an adjacent control point can be created dynamically for the adjacent node.

Note: This operand is ignored if CPNAME is coded on the physical unit.

DYNADJCP=NO
Indicates that an adjacent control point cannot be dynamically created.

DYNADJCP=YES
Indicates that an adjacent control point can be dynamically created.

DYNLU

```
      ┌─DYNLU=value_of_dynlu_start_option──────┐
►►────┤                                        ├──────────────────────────────►◄
      │              ┌─value_of_dynlu_start_option─┐
      └─DYNLU=───────┤                             ├──┘
                     ├─NO────────────────────────────┤
                     └─YES───────────────────────────┘
      ┌─DYNLU=YES─┐   (1)
      │           ├──────────────────────────────────┘
      └─DYNLU=YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statements: *GROUP, LINE, PU*
dependencies: *VTAM only; PUTYPE=1 or 2*

Specifies whether dynamic allocation of CDRSC definitions is supported. If you do not code this operand, the default is the value you coded on the DYNLU start option.

This operand is valid only when PUTYPE=1 or PUTYPE=2.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, a resource that was dynamically defined as a CDRSC cannot use that PU to connect to a session.

DYNLU=NO
Specifies that dynamic allocation of CDRSC definitions is not supported.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, an independent logical unit that was dynamically defined using CDRSC definitions over another adjacent link station cannot use that particular PU to connect to a session.

DYNLU=YES
Specifies that dynamic allocation of CDRSC definitions is supported.

For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

DYNPU

```
►►─┬─DYNPU=NO──────────┬──────────────────────────────────────────►◄
   │                   │
   └─DYNPU=─┬─NO──┬────┘
            └─YES─┘
```

statements:	*GROUP*
dependencies:	*none*

Specifies whether a PU is to be dynamically allocated when the calling PU cannot be identified during a switched call-in operation. DYNPU applies to APPN and subarea PUs.

A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:
- MAXOUT=8
- ANS=CONT
- DISC=(YES,F)
- DYNADJCP=YES
- CPCP=YES

DYNPU=NO
Specifies that dynamic PUs are not to be used.

DYNPU=YES
Specifies that dynamic PUs can be used.

DYNPUPFX

```
   ┌─DYNPUPFX=CN──────┐
►►─┤                  ├──────────────────────────────────────────────────►◄
   └─DYNPUPFX=string──┘
```

statements: *GROUP*
dependencies: *DYNPU=YES*

Specifies a 2-character identifier used to create an 8-byte PU name when a dynamic PU is created for the link station. The two characters are used at the beginning of the PU name.

Note: This operand is valid only when DYNPU is YES.

DYNPUPFX=CN
Indicates the beginning of the PU name is CN. The default value enables you to quickly identify the name of a dynamically defined PU.

DYNPUPFX=*string*
Allows you to create your own identifier. Specify two characters.

If you create your own identifier instead of using the default, follow the naming conventions for VTAM as stated in “Coding VTAM Definitions” on page 8.

ECLTYPE

```
►►─┤ECLTYPE├──────────────────────────────────────────────────────────────►◄
```

statements: *GROUP*
dependencies: *VTAM and NCP*

VTAM uses the ECLTYPE operand and DIAL=YES to detect lines used for peripheral devices that are connected through NCP/Token-Ring interface (NTRI) support.

If you have NCPs connected over a token ring, you can use ECLTYPE to define a private frame-relay network over the token ring without requiring additional physical lines.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the ECLTYPE operand.

ENCR

```
   ┌─ENCR=NONE──────┐
►►─┤                ├──────────────────────────────────────────────────►◄
   │          ┌─NONE─┐
   └─ENCR=────┤      ├
              ├─OPT──┤
              ├─REQD─┤
              └─SEL──┘
```

statements: *GROUP, LINE, PU, LU*
dependencies: *VTAM only*

Specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

ENCR=NONE
Specifies that this logical unit cannot engage in cryptographic sessions.

ENCR=OPT
Specifies that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from this logical unit. If you code ENCR=REQD, no session can be established with this logical unit unless the host with which it is associated can handle cryptographic sessions.

ENCR=SEL
Has no meaning for logical units. If you code ENCR=SEL, ENCR=OPT is used instead.

ENCRTYPE

```
   ┌─ENCRTYPE=DES───────┐
►►─┤                    ├──────────────────────────────────────────────►◄
   └─ENCRTYPE=TDES24────┘
```

statements: *GROUP, LINE, PU, LU*
dependencies: *none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

Note: The ENCRTYPE value coded on the GROUP, PU, or LINE will be sifted down to the appropriate resource (LU).

The SYSDEFed ENCRTYPE of a LU can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications*

Server: SNA Operation for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to "MODEENT Macroinstruction" on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements:	*GROUP, LINE, PU, LU*
dependencies:	*VTAM only; TSO sessions only*

Determines how session establishment is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The PLU acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU's application program, only a single BIND is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) that establish sessions with TSO.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FEATUR2

```
►► ┌─FEATUR2=(MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────┐
   │                                                                      ├──►◄
   └─FEATUR2=(─┬─MODEL1──┬─┬─NOEDATS──┬─┬─NOPRINTR──┬─┬─NOSELPEN──┬─)──────┘
               └─MODEL2──┘ └─EDATS────┘ └─PRINTR────┘ └─SELPEN────┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only*

Describes the machine features for a specific link-attached non-SNA terminal. Refer to the *Planning for NetView, NCP, and VTAM* for a description of start-stop terminal planning considerations.

When coding the FEATUR2 operand on the PU and LU definition statements refer to the *Network Terminal Option Installation* for more information.

FEATUR2=DUALCSE

FEATUR2=LOWERCSE
Specifies how VTAM sends alphabetical characters coded with the TEXT operand on a USSMSG macroinstruction to a non-SNA terminal over the SSCP-LU session. This value does not affect non-alphabetical characters or any characters coded on the BUFFER operand of a USSMSG macroinstruction.

Code LOWERCSE to indicate that alphabetical characters are sent to the terminal over the SSCP-LU session in lowercase. Code DUALCSE to indicate that VTAM sends all characters as they are coded in the USSMSG macroinstruction.

FEATUR2=EDATS

FEATUR2=NOEDATS
Tells VTAM whether this terminal has the extended data stream feature. You cannot use this operand for terminals attached by SDLC lines.

Coding EDATS for a BSC line will cause the NCP to place the line in transparent text mode.

FEATUR2=MODEL1

FEATUR2=MODEL2
Identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

This information is available to an application program as part of the device characteristics pertaining to this terminal. You can obtain the characteristics using the INQUIRE macroinstruction. For more information about using the INQUIRE macroinstruction, see *OS/390 eNetwork Communications Server: SNA Programming*.

FEATUR2=NOPRINTR

FEATUR2=PRINTR
Specifies whether this terminal has an attached IBM 3284 Model 3 printer. This

operand is valid only if you code TERM=3275 or if TERM=3275 is assumed by NCP definition statement sequencing for this statement.

FEATUR2=NOSELPEN

FEATUR2=SELPEN
Specifies whether this terminal supports a selector pen.

FRELAY

```
►►──┬─FRELAY=(PHYSICAL──┬──────────────┬──)──┬──────────────────────────────────►◄
    │                   ├─,PERIPHERAL──┤     │
    │                   ├─,SUBAREA─────┤     │
    │                   └─,ANY─────────┘     │
    └─FRELAY=(LOGICAL───┬──────────────┬──)──┘
                        ├─,PERIPHERAL──┤
                        ├─,SUBAREA─────┤
                        └─,IP──────────┘
```

statements:	*GROUP*
dependencies:	*VTAM and NCP; PU type 1 only*

Indicates the type of frame-relay group and the types of logical lines supported.

The names of PUs under the GROUP and LINE statements in a frame-relay group must be unique.

In addition, every frame handler PU under a physical frame-relay line must be put in a FRSESET definition statement.

When a frame-relay PU is dynamically defined but not included in a FRSESET definition statement, the resource is validated by VTAM, but will not be activated because one of the requirements for defining a frame-relay PU is that the PU needs to be included in a FRSESET definition. Frame-relay PUs not included in a FRSESET will generate messages informing you of the requirement. Update the definition list to include a FRSESET definition for the frame-relay PUs that are not included in one, replace the existing definition list with the revised one, and issue a V NET,ACT,UPDATE=ALL.

FRELAY=PHYSICAL|PHY
Indicates frame-relay physical lines. The types of logical lines supported by the physical line is determined by the second suboperand (PER, SUB, ANY).

- Code FRELAY=(PHYSICAL,SUBAREA|SUB) to define a physical line that supports logical subarea lines only.
- Code FRELAY=(PHYSICAL,PERIPHERAL|PER) to define a physical line that supports logical peripheral lines only.
- Code FRELAY=(PHYSICAL,ANY) to define a physical line that supports both logical peripheral and logical subarea lines.

If you code FRELAY=PHYSICAL and do not specify SUBAREA, PERIPHERAL, or ANY, the default for the second suboperand depends on the line address specified by ADDRESS on the LINE definition statement. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information about the ADDRESS operand.

- If the line address is 1–895 then the default is (PHYSICAL,SUBAREA).

- If the line address is 1024–1039 or 2112–2750, then the default is (PHYSICAL,ANY).

FRELAY=LOGICAL
Defines the logical line group.

- Code FRELAY=LOGICAL or FRELAY=(LOGICAL,SUBAREA|SUB) to define a logical subarea line group. The PU definition statement specified by the PHYSRSC operand must be included in a line group with FRELAY=PHYSICAL.
- Code FRELAY=(LOGICAL,PERIPHERAL|PER) to define a logical peripheral line group. If a pu_name is coded on the PHYSRSC keyword (not PHYSRSC=NONE), then the PU definition statement specified by the PHYSRSC operand must be included in a line group with FRELAY=PHYSICAL.
- Code FRELAY=(LOGICAL,IP) to define a logical line group to be used for IP connections over a switched physical line. The PU definition statement specified by the PHYSRSC operand must be included in a line group with FRELAY=PHYSICAL and DIAL=YES.

GPOLL

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └GPOLL=polling_character┘
```

statements: *CLUSTER*
dependencies: *VTAM and NCP*

GPOLL (general polling) must be coded for BSC 3270 clusters (that is, 3271 and 3275); VTAM does not support specific polling.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the GPOLL operand.

GP3174

```
    ┌GP3174=NO──────────────┐
►►──┼───────────────────────┼──────────────────────────────────────────►◄
    │          ┌NO──────┐   │
    └GP3174=───┼────────┼───┘
               └group_address┘
```

statements: *GROUP, LINE, PU*
dependencies: *VTAM and NCP*

Specifies that a poll to an IBM 3174 gateway controller can be a group poll for any station on the IBM 3174, including the IBM 3174 and the stations attached to the token ring.

group address indicates the hexadecimal group poll address and can be any two hexadecimal digits except X'00' and X'FF'.

You can change the value of GP3174 for SDLC nonswitched PUs by dynamically adding the PUs using the DR ADD definition statement.

GWCTL

```
   ┌─GWCTL=SHR──────┐
►►─┤                ├──────────────────────────────────────────────►◄
   │          ┌─SHR─┐│
   └─GWCTL=───┤     ├┘
              └─ONLY┘
```

statements: *PCCU*
dependencies: *SNI only*

Determines whether the host's SSCP does all the gateway control functions during LU-LU session setup. GWCTL determines the SSCP's role for only the gateway NCP being defined, and applies only if the SSCP is on the session setup path.

GWCTL=ONLY
Specifies that only one SSCP controls the gateway NCP.

If the session setup path crosses three networks within a single gateway NCP, and if one or more gateway SSCPs are in session with the gateway NCP, the gateway SSCP in the center network must be designated to perform all of the gateway functions. Code GWCTL=ONLY on the PCCU statement for the gateway SSCP in the center network.

GWCTL=SHR
Indicates control of the gateway NCP is shared. If one gateway SSCP cannot perform the gateway control function, the other SSCP performs the function.

For an inactive NCP, the VARY ACT command processes all operands coded on the PCCU definition statement. The VARY ACQ command, however, processes only the GWCTL operand, regardless of what is coded on the PCCU definition statement.

For more information, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

HPR

```
                (1)
   ┌─HPR=YES────────┐
►►─┤                ├──────────────────────────────────────────────►◄
   │          ┌─NO──┐│
   └─HPR=─────┤     ├┘
              └─YES─┘
```

Notes:

1 HPR defaults to NO when the HPR start option is set to NONE, (RTP,NONE), or (ANR,NONE).

statements: *GROUP, LINE, PU*
dependencies: *Type 2.1;*
NCP V7R3 required

Specifies whether high performance routing (HPR) support is enabled for a particular GROUP, LINE, or PU.

HPR=YES
Indicates that this GROUP, LINE, or PU provides HPR support.

HPR=NO
Indicates that this GROUP, LINE, or PU does not provide HPR support.

INBFRS

```
►►──INBFRS=number_of_buffers──────────────────────────────────────────►◄
```

statements:	*HOST*
dependencies:	*none*
range:	*1–255*

Defines the number of buffers to be allocated by an NCP to receive data from a host.

INCLUD0E

```
                ┌─INCLUD0E=NO──────┐
►►──────────────┤                  ├──────────────────────────────────►◄
                └─INCLUD0E=──┬─NO──┬┘
                             └─YES─┘
```

statements:	*PU*
dependencies:	*none*

Specifies whether VTAM provides LU names to the logical units defined under this PU. The names are provided by passing a CV0E, Network Name control vector on the ACTLU requests.

INCLUD0E=NO
Indicates that VTAM will not provide LU names to the logical units defined under this PU.

INCLUD0E=YES
Indicates that VTAM will provide LU names to the logical units defined under this PU.

INITEST

```
                ┌─INITEST=NO───────┐
►►──────────────┤                  ├──────────────────────────────────►◄
                │            ┌─NO──┐
                └─INITEST=───┤     ├┘
                             └─YES─┘
```

statements:	*PCCU*
dependencies:	*see notes*

Specifies whether VTAM loads the initial test routine, which is a diagnostic routine for a channel-attached communication controller. This routine checks the communication controller for any machine malfunctions before VTAM loads the NCP into it.

INITEST=NO
Specifies that VTAM will not load the initial test routine.

INITEST=YES
Specifies that VTAM will load the initial test routine.

Notes:

1. Applies only to channel-attached IBM 3705 Communication Controllers.
2. If you code INITEST=YES for any communication controller other than an IBM 3705, the load fails.
3. If you code INITEST=YES for an NCP that controls a link-attached IBM 3705 Communication Controller, VTAM ignores it.

INNPORT

```
                ┌─INNPORT=NO──────┐
►►──────────────┤                 ├──────────────────────────────────────────►◄
                │          ┌─NO─┐ │
                └─INNPORT=─┤    ├─┘
                           └─YES─┘
```

statements:	*PU*
dependencies:	*VTAM only*

Specifies whether this physical unit supports intermediate network node (INN) traffic. This operand is not necessary for PU type 4; it is only for PU type 1 used for INN traffic.

INNPORT=NO
INNPORT=NO indicates that the physical unit is either not an INN port or is not a port.

INNPORT=YES
INNPORT=YES means that the physical unit represents a port that supports INN traffic. If you specify INNPORT=YES, this physical unit should not be deactivated when CDLINK=ACT on either the HALT or VARY INACT commands.

For frame-relay PUs, code INNPORT=YES.

This operand is created with its default value by the NCP generation process. Code INNPORT=YES only if network definition facility (NDF) changes the default value during the generation process.

Note that in certain definitions, NDF sets INNPORT to YES. When any of the following operands is specified on the GROUP definition statement of the NCP generation definition, NDF adds INNPORT=YES to all PU definition statements within that GROUP statement when the PU statements are written to the NEWDEFN file:

- ECLTYPE=(PHYSICAL, SUBAREA)
- ECLTYPE=(PHYSICAL,ANY)
- FRELAY=PHYSICAL
- ETHERNET=PHYSICAL

ISTATUS

```
   ┌─ISTATUS=ACTIVE──────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   │                ┌─ACTIVE─┐│
   └─ISTATUS=───────┤        ├┘
                    └─INACTIVE┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only*

Determines whether this minor node is to be initially active when the NCP major node to which it belongs is first activated, either due to a cold start of VTAM, or, after VTAM is started, by issuing the VTAM VARY ACT,COLD command for the NCP major node. For logical units, ISTATUS is also effective whenever the associated physical unit is activated.

When coding the ISTATUS operand, consider these special cases:

- For a line with TYPE=PEP, ISTATUS does not apply and is ignored if you code it.
- For a line with USE=NCP, the line is initially active.
- For a line with USE=EP, the line is initially inactive.

Code ISTATUS=INACTIVE for all start-stop lines supported by Network Terminal Option (NTO).

The operator can override the ISTATUS specification using the SCOPE operand on the VARY ACT command.

If LNCTL=CA (on GROUP), you must specify ISTATUS=INACTIVE on the LINE definition statement if PUTYPE=5 on the PU definition statement.

ISTATUS=ACTIVE
Indicates that the minor node is automatically activated when its major node is activated.

ISTATUS=INACTIVE
Indicates that the minor node is not automatically activated when its major node is activated.

LIMRES

```
   ┌─LIMRES=NO───────────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   │                ┌─NO─────┐│
   │ └─LIMRES=──────┤        ├┘
   │                └─YES────┘
   │ ┌─LIMRES=NO──┐   (1)
   └─┤            ├
     └─LIMRES=NO──┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

statements: *GROUP, LINE, PU*

dependencies: *VTAM; LIMQSINT; LU 6.2 only; PUTYPE=1 or 2*

Specifies whether a group of lines, a line, or a physical unit is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes any sessions that traverse the resource to be deactivated if no conversation is active. A session can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

If you do not code a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

You can also use LIMRES in conjunction with the LIMINTCP start option to specify the interval to retain a free session using a limited resource for some network management flows (excluding flows using the APPN control point sessions or dependent LU server sessions). If you do not code a value for the LIMINTCP start option, resources are not considered to be limited resources, and the sessions will remain active even after all conversations have ended.

Notes:

1. For NCP-defined resources, the NCP sets the limited resource indicator. If the NCP is pre-V5R4, then the limited resource indicator is not set by the NCP, even if VTAM definitions indicate that a limited resource is defined.
2. This operand is not valid when PUTYPE=4 or PUTYPE=5.
3. You should code LIMRES only for leased lines in the NCP major node. For switched lines used by the NCP major node, code LIMRES in the switched major node.

LINEAUT

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └─LINEAUT=YES──┘
```

statements: *GROUP, LINE*
dependencies: *VTAM and NCP*

Specifies that VTAM is to consider every line in a group as an auto-call line. If you omit this operand, but LINEADD=NONE and DIAL=YES, then LINEAUT=YES is assumed by default.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the ECLTYPE operand.

LLERP

```
    ┌─LLERP=NOTPREF──────┐
    ├─LLERP=REQUIRED─────┤   (1)
►►──┼────────────────────┼──────────────────────────────────────────────►◄
    └─LLERP=──┬─REQUIRED─┬─┘
              └─NOTPREF──┘
```

Notes:

1 For type 2.1, the default is LLERP=REQUIRED. For SDLC, X.25, and channels links, the default is required. Otherwise, the default is LLERP=NOTPREF.

statements:	*PU*
dependencies:	*HPR=YES*

Specifies the link-level error recovery procedures (LLERP) preference for HPR connections.

LLERP=REQUIRED
Specifies that link-level error recovery procedures are required.

LLERP=NOTPREF
Specifies that link-level error recovery procedures are required by this PU only when the adjacent link station requires it.

LMODADR

```
►►──┬──────────────
```

Specifies the type of line control used by the lines in the group being defined. For more details, see the *NCP, SSP, and EP Resource Definition Reference*.

LNCTL=BSC
Indicates that the definition statements following this GROUP definition statement use the BSC line protocol.

LNCTL=CA
Indicates that the definition statements following this GROUP definition statement identify intermediate routing node and boundary node channel definitions.

LNCTL=SDLC
Indicates that the definition statements following this GROUP definition statement use the SDLC line protocol.

LNCTL=SS
Indicates that the definition statements following this GROUP definition statement use the SS line protocol.

LNCTL=USER
Indicates that either the user or NCP provides the level-5 code. The LEVEL2 and LEVEL3 operands must be specified if LNCTL=USER is coded.

LOADFROM

```
                  ┌─LOADFROM=HOST──────────┐
►►──────────────────┼────────────────────────┼──────────────────────────────────────►◄
                  │                        │
                  │            ┌─HOST──┐   │
                  └─LOADFROM=──┼───────┼───┘
                               └─EXTERNAL─┘
```

statements:	*PCCU*
dependencies:	*see notes*

Specifies whether VTAM loads the NCP module from the host or from the communication controller external disk storage.

LOADFROM=HOST
Loads the NCP from the host.

LOADFROM=EXTERNAL
Loads the NCP from the 3720 or 3745 communication controller.

Notes:

1. You cannot specify both SAVEMOD=YES and LOADFROM=EXTERNAL on the PCCU definition statement. If you specify LOADFROM=EXTERNAL, you should take the default value for SAVEMOD.
2. You cannot specify DUMPLOAD=YES, LOADFROM=HOST, and SAVEMOD=NO all together. If you specify LOADFROM=HOST and SAVEMOD=NO, you should take the default value for DUMPLOAD.
3. You cannot specify SAVEMOD=YES on the VARY ACT command and LOADFROM=EXTERNAL on this PCCU definition statement. If you do, VTAM does not use LOADFROM=EXTERNAL on the initial load.

LOADLIB

```
►►──LOADLIB=ddname──────────────────────────────────────────────────►◄
```

statements: *BUILD*
dependencies: *none*

Specifies the data set or file in which the NCP resides.

Name the data set on a DD statement in the VTAM start procedure.

LOADSTA

```
►►──┬──────────────────────────┬──────────────────────────────────►◄
    └─LOADSTA=link_station_name─┘
```

statements: *PCCU*
dependencies: *none*

Specifies an adjacent link station (channel- or link-attached) that is used to load the NCP. If you do not code LOADSTA on the PCCU definition statement or specify it in the VTAM VARY ACT command, VTAM chooses a default link station whenever it needs one. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how VTAM chooses a default link station.

For a channel-attached communication controller, if the link station name is not defined (in the channel attached major node), VTAM assigns a link station name by taking the device number and adding “-S”. If the device number is 3-digits, a leading zero is added to the name. For example, the device number 0C2 has the link station name of 00C2-S.

You can designate the same link station for both dumping and loading.

For more information on link stations and use of the VTAM VARY ACT command, refer to *OS/390 SecureWay Communications Server: SNA Operation*.

LOCADD

```
►►──LOCADD=address──────────────────────────────────────────────────►◄
```

statements: *LINE*
dependencies: *VTAM and NCP*

Specifies the locally administered address for the token-ring interface coupler (TIC). The value of LOCADD must specify an address among all the local and remote token-rings to which this TIC is connected. LOCADD is required when you specify physical resources on the GROUP definition statement.

Use the following format when coding LOCADD:

```
LOCADD=cccccccccccc
```

where:

cccccccccccc
is the last 6 bytes of the terminal's ring-station address (the first digit must be in the range 4–7). The address must consist of 12 hexadecimal digits that range from X'4000 0000 0000' to X'7FFF FFFF FFFF' using the following rules.

- If bytes 0 and 1 are equal to either X'4000' or X'7FFF' then bytes 2 through 5 must be unique on the network.
- If bytes 0 and 1 are in the address range X'4001' through X'7FFE' then bytes 0 through 5 must be unique on the network.

A locally administered medium access control (MAC) address coded for LOCADD cannot have the same value as that specified on the ADDR operand on the PU definition statement. Refer to the *NCP, SSP, and EP Resource Definition Reference* for additional information.

LOCADDR

```
►►──LOCADDR=address──────────────────────────────────────────────────►◄
```

statements:	*LU*
dependencies:	*VTAM and NCP*
range:	*0–255 PU type 2 and type 2.1 0–63 PU type 1*

Specifies the logical unit's local address at the associated physical unit. Specify the addresses in ascending order. LOCADDR=0 specifies an independent LU.

An LU definition statement is not required for every possible local address, and LOCADDR values need not be consecutive. Unused local addresses smaller than the largest local address at a station are not assigned network resources.

Notes:

1. The value of LOCADDR depends on the requirements of the device being defined. Consult the appropriate component description manual for these restrictions; for example, logical units associated with a 3279 or 3276 control unit must have a LOCADDR in the range 2–33.
2. Although you can use this operand to define independent LUs, it is recommended that you either use dynamic definition or predefine the independent LUs as CDRSCs.
3. VTAM cannot recognize a dynamically changed LOCADDR on the first activation of an NCP, even when VARY ACT,UPDATE=ALL is used. (Warm starts are considered initial activations.) However, on changes to LOCADDR after the first activation, VTAM can compare the value in the definition statement to the value previously coded and recognize the change.

LOGAPPL

```
►►──┬──────────────────────────┬──────────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu─┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies:	*VTAM only; NetView or TSO*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

For independent logical units, the LOGAPPL operand establishes a session with the primary logical unit designated in the LOGAPPL operand if a session does not currently exist with this logical unit. The session is established regardless of whether sessions with other logical units exist. The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

Automatic logon LU-LU sessions are not redriven after CP-CP outages even though the CP-CP sessions might redrive.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─LOGTAB=interpret_table_name─┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only; dependent LUs only*

Specifies the name of an interpret table that VTAM uses to associate a logon request with the name of an application program or routine to which this logon is sent. VTAM uses the interpret table to interpret the name received from an initiation request. For more information on using interpret tables in conjunction with USS tables, see “Unformatted System Services (USS) Tables” on page 731.

LOGTAB is valid for dependent logical units only.

LPDA

```
►►──┬─LPDA=ALLOW──────────┬──────────────────────────────────────────────►◄
    │                     │
    └─LPDA=─┬─ALLOW─┬─────┘
            └─BLOCK─┘
```

statements: *PU*
dependencies: *VTAM and NCP*

Specifies whether Link Problem Determination Aid (LPDA) tests can be run for the physical unit.

Note: Do not specify this keyword if PUTYPE=4 has been specified.

LPDA=ALLOW
Indicates LPDA test will run for the PU.

LPDA=BLOCK
Indicates no LPDA test will run for the PU.

LUDR

```
►►─┬─LUDR=YES──────────┬──────────────────────────────────────────►◄
   │                   │
   └─LUDR=─┬─YES─┬─────┘
           └─NO──┘
```

statements:	*GROUP, LINE, PU, LU*
dependencies:	*VTAM and NCP*

Specifies whether an LU is eligible for dynamic reconfiguration.

This operand is ignored for independent LUs.

If you use sifting for LUDR and use VARY ACT,UPDATE=ALL to move an LU, make sure the sifted value is the same before and after any moves. If LUDR gets sifted differently as the result of a move and its value is no longer the same, VTAM and NCP will have mismatching LUDR values on the next initial activation of the NCP.

LUDR=NO
Indicates that VTAM rejects any attempt to move or delete the LU.

LUDR=NO is not valid if PUDR=YES. However, by using DRDS definitions you can create an LU with LUDR=NO under a PU with PUDR=YES, and DR operations on the LU will fail, but DR operations on the PU ignore the LUDR=NO value.

LUDR=YES
Indicates that the LU can be dynamically deleted or moved. Valid only if dynamic reconfiguration is supported for this LU.

LUGROUP

```
►►─┬──────────────────────────┬──────────────────────────────────►◄
   └─LUGROUP=lu_group_name────┘
```

statements:	*PU*
dependencies:	*VTAM only*

Specifies the name of the model LU group that VTAM uses to select a model LU definition when dynamically defining an LU. This operand also indicates that the device supports dynamic definition of LUs.

LUSEED

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─LUSEED=lu_pattern_name─┘
```

statements: *PU*
dependencies: *VTAM only; LUGROUP*

Provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU. LUSEED is valid only when you code the LUGROUP operand.

If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- The first character must be either alphabetical or one of the national characters (@ or $).
- The remaining characters can be either alphabetical, numerical, or national.
- There must be either two or three contiguous national # characters in the name. These characters represent holders, in which the logical unit's local address is substituted when an LU's name is generated. If you specify three # characters, the LU's logical address is substituted in decimal for the # characters. If you specify two # characters, the LU's logical address is substituted in hexadecimal for the # characters.

When used with a user-written SDDLU exit routine, this operand is optional and there are no restrictions on its format. For more information on the SDDLU exit, see *OS/390 SecureWay Communications Server: SNA Customization*.

MAXBFRU

```
►►──MAXBFRU=number_of_buffers──────────────────────────────────────────────►◄
```

statements: *HOST*
dependencies: *none*
range: *1–255*

Specifies the number of buffers to be allocated by VTAM to receive data from an NCP. The value coded for MAXBFRU must satisfy the condition that MAXBFRU times the IOBUF size is less than 65535.

If you define LNCTL=NCP in a channel-attachment major node that is activated before the NCP major node, ensure that the MAXBFRU value in the NCP major node matches the MAXBFRU value in the channel-attachment major node. Because the MAXBFRU value in the channel-attachment major node is processed first, an error message is issued if the MAXBFRU value in the NCP major node differs from the value in the channel-attachment major node.

Note: If you are defining the channel to the NCP in a channel-attachment major node, VTAM might ignore this operand when the NCP is activated by a CMC host. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on CMC configurations.

MAXDATA

```
►►─┬─MAXDATA=65535──────────┬──────────────────────────────────────────►◄
   └─MAXDATA=number_of_bytes─┘
```

statements: *PCCU*
dependencies: *none*

Specifies the maximum number of bytes that the NCP can receive in one PIU or PIU segment, including the transmission header (TH) and the request/response header (RH).

To optimize the processing of a MODIFY DUMP, OPTION=DYNAMIC command, MAXDATA should be at least 2082 (2048 plus 34, the size of the PIU header).

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about choosing a value for MAXDATA.

Note: If you plan to define the channel to the NCP in a channel-attachment major node, this operand is replaced by the MAXDATA operand on the PU definition statement for the channel-attachment major node. See “MAXDATA” on page 76 for a description of the MAXDATA operand in a channel-attachment major node.

MAXOUT

```
►►─┬─MAXOUT=8──────────────┬──────────────────────────────────────────►◄
   └─MAXOUT=number_of_pius─┘
```

statements: *GROUP, LINE, PU*
dependencies: *none*

Specifies the maximum number of unacknowledged information frames that NCP can have outstanding for a PU before requesting a response from the PU.

For frame-relay PUs, the default is 8.

MAXSESS

```
►►──┬─MAXSESS=max_number_LU-LU_sessions─┬──────────────────────────────►◄
```

statements: *GROUP, LINE, PU, LU*
dependencies: *VTAM and NCP; independent LU*
range: *1–65535*

Specifies the maximum number of concurrent LU-LU sessions in which an independent LU can participate. The MAXSESS value applies to each connection the LU is using to establish LU-LU sessions. For example, if MAXSESS=2 was

specified for LUA and LUA appears on three PUs, LUA can have two sessions on each of the three PUs for a total of six sessions.

If MAXSESS is not coded, the value specified on the BUILD definition statement will be used. Refer to the *NCP, SSP, and EP Resource Definition Reference* for additional information about this operand.

MAXSUBA

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └MAXSUBA=n─────┘
```

statements:	*BUILD, NETWORK*
dependencies:	*ENA*
range:	*1–255*

Specifies the highest subarea value that can be assigned within the network for communicating with a subarea node that does not support extended network addressing (ENA).

This operand is required if any host or NCP in the network does not support ENA. All nodes must specify the same value.

For more information, see “MAXSUBA” on page 606. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for considerations when coding this operand.

MDLENT

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └MDLENT=model_name_table_entry_name┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies:	*VTAM only; MDLTAB*

Specifies the name of the model name table entry to be used for a logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB.

MDLTAB

```
►►──┬──────────────────────────┬──────────────────────────────────────►◄
    └MDLTAB=model_name_table_name┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies:	*VTAM only*

Specifies the name of the model name table to be used for a logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

The model name is only passed when VTAM initiates the session with the application program, for example, only when the LOGAPPL = parameter is also coded.

If you omit the LOGAPPL operand or the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MDUMPDS

```
►►─┬─MDUMPDS=value_coded_on_dumpds─┬──────────────────────────────────────►◄
   │                               │
   └─MDUMPDS=moss_dump_name────────┘
```

statements: *PCCU*
dependencies: *none*

Specifies one of three operands (the others are DUMPDS and CDUMPDS) that name dump files for this communication controller. MDUMPDS applies only to 3720, 3725, and 3745 communication controllers.

This operand is optional. If it is coded, it names the file that contains maintenance and operator subsystem (MOSS) dumps. If it is not coded, the MOSS dumps are contained in the file named on the DUMPDS operand. For more information on DUMPDS, see “DUMPDS” on page 338.

dump_name must be on a DD statement, which defines the file that is to contain the data from a storage dump of a communication controller.

MODETAB

```
►►─┬─MODETAB=ISTINCLM──────────────┬──────────────────────────────────────►◄
   │                               │
   └─MODETAB=logon_mode_table_name─┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only*

Specifies the name of a logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►──┬──────┬──────────────────────────────►◄  ►►──name──────────────────────────►◄
    └name──┘
```

statements:	PCCU, BUILD, HOST, NETWORK, FRSESET	*statements:*	GROUP, LINE, PU, LU, CLUSTER, TERMINAL

Assigns a name to the resource being defined by the definition statement.

On GROUP, LINE, PU, LU, CLUSTER, TERMINAL: Provides the required network name for the resource being defined by the definition statement.

On PCCU: Provides an optional name for this major node. Although *name* is optional, it is recommended that you code one because the node can then be identified by name when messages are issued regarding the node's status.

On BUILD: Provides an optional name for the definition statement.

NATIVE

```
►►──┬──────────────────┬──────────────────────────────────────────────────────►◄
    └NATIVE=──┬──NO──┬─┘
              └──YES─┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only; BN=YES*

Specifies whether this link station represents a connection to a native node.

If NATIVE is not specified on the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

If you code NATIVE on both the ADJCP and PU definition statement, the values must match.

For transmission groups (TGs) between nodes, the value specified or negotiated for NATIVE must match. If a TG is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the TG attempting activation is not allowed to become active. If the activation is for a virtual-route-based transmission group then the SSCP-SSCP session will also fail.

NETID

```
►►──NETID=network_id──────────────────────►◄  ►►──┬──────────────────────┬──────────────►◄
                                                  └──NETID=network_id────┘
```

statements:	BUILD	*statements:*	*PCCU, GROUP, HOST, NETWORK PU*
dependency:	VTAM and NCP	*dependency:*	VTAM and NCP; SNI only; COPIES

Specifies 1–8 character network identifier.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

NETID=(*network_id***)**

On PCCU: (*network_id*) specifies the 1–8 character name of the network of the host represented by this PCCU definition statement. VTAM uses this name along with the SUBAREA operand to determine which PCCU definition statement it should process in the NCP major node.

On BUILD: (*network_id*) specifies the network name of the host coded on the NETID start option. This operand is required for NCP V4R3 and later releases.

On HOST: (*network_id*) specifies a 1–8 character name that identifies the network containing the corresponding host node. It is used along with the SUBAREA operand to determine which HOST definition statement the VTAM host should process. This operand is optional.

On NETWORK: (*network_id*) specifies the 1–8 character network identifier of the nonnative network in which the gateway SSCP resides. If the value for NETID matches the value of VTAM's NETID start option, VTAM uses the MAXSUBA value on this statement as the maximum subarea value, and the SUBAREA value is the actual subarea for the gateway NCP representation in the VTAM network.

On PU: (*network_id*) specifies a 1–8 character network identifier. For type 4 and 5 physical units, *network_id* identifies the network of the subarea serviced by the line. NETID is used only in gateway NCPs, and is required for a connection to another network.

Note: For switched subarea connections, all lines within the same group must be defined with the same *network_id*.

For type 2 physical units, NETID is used in conjunction with the XNETALS start option in support of nonnative network connection. See "XNETALS" on page 657 for information about XNETALS.

If you specify NETID on the PU definition statement, VTAM ensures that when the physical unit is active, the connecting resource is within the network specified by NETID. If you omit NETID, VTAM dynamically processes the network ID when a connection is established. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM's network ID.

If dynamic dial-out connections are required (that is, a session request drives the dial), NETID must be predefined. Otherwise, VTAM might not be able to determine the connecting network ID and the resultant sessions might fail. If leased connections are not established, and if NETID is not predefined, directory requests could fail because of VTAM's inability to determine the resource's actual network.

NEWNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─NEWNAME=ncp_load_module──┘
```

statements: *BUILD*
dependencies: *none*

Specifies 1–7 characters for the NCP load module name. It must be a unique NCP name across the network. This name must match the member name for the NCP source module in both VTAMLST and the LOADLIB specified.

If you do not code PUNAME, NEWNAME also represents the resource name for the NCP physical unit (PU Type 4).

If you code NEWNAME, and LOADMOD is specified on the VARY ACT command, they must match. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on LOADMOD.

NEWPATH

```
►►──┬──────────────────────────────────────────┬──────────────────────────►◄
    └─NEWPATH=──┬─name──────────────────────┬──┘
                ├─(name)────────────────────┤
                ├─(name1,name2)─────────────┤
                └─(name1,name2,name3)───────┘
```

statements: *PCCU*
dependencies: *none*

Specifies the dynamic path update member names in the VTAM definition library. When the NCP is successfully activated, VTAM processes and sends path table update specifications to the NCP.

Up to three member names can be specified; if more than three member names are specified on the PCCU definition statement, only the first three are used.

VTAM sends the path table update specifications over the SSCP-PU session before any links, other than the link for the SSCP-PU session, are activated. If you attempt

to change or delete the explicit route used for the SSCP-PU session, the NCP rejects the request (because the explicit route is currently operative), and VTAM displays a warning message.

Any NEWPATH operand specification on the PCCU definition statement is nullified if the NEWPATH operand is specified in the VARY ACT command for the NCP. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information on the VARY ACT command.

When a dynamic path update member is processed as a result of a NEWPATH operand on the PCCU definition statement, path update specification sets for other subareas, if any, are not applied.

NN

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └─NN=──┬─NO──┬─┘
           └─YES─┘
```

statements: *GROUP, LINE, PU*
dependencies: *APPN only*

Specifies whether the adjacent node is expected to be a network node. If a value for the NN operand is specified, it will be validated when a connection is attempted to the adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If the NN operand is not specified, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

OWNER

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─OWNER=──┬─owner_name──────────┬─┘
              └─(──owner_name──────)─┘
```

statements: *PCCU, GROUP, LINE, PU*
dependencies: *VTAM only; on PU, PUTYPE=1 or 2*

PCCU: Defines an identifier for the VTAM defined by the PCCU statement. The identifier is used to associate a host VTAM with the resources it controls. You can use this operand to divide the resources among the hosts that are in session with an NCP.

If you code OWNER, the activating host owns the resources that do not have the OWNER operand specified or have OWNER coded with a value that matches *owner name*. If you do not specify OWNER, all resources are owned by the activating host, whether the OWNER operand is specified on the resources.

GROUP, LINE, PU: Associates the resources in a communication controller with the VTAM that controls these resources. OWNER associates those resources with the PCCU definition statement that has the same OWNER value. A resource can be associated with several VTAMs. If you do not specify the OWNER operand, the resource can be owned by any host.

Note that for the PU definition statement, OWNER is valid only when PUTYPE=1 or 2.

Notes:

1. The OWNER operand is valid on the LINE or PU definition statement for nonswitched lines. It is also valid on the LINE definition statement for switched lines.
2. If you do not specify the OWNER operand on the PCCU definition statement, the OWNER operand coded on a GROUP, LINE or PU definition statement is ignored.
3. If the OWNER operand on a GROUP, LINE or PU definition statement does not match the OWNER operand on the PCCU definition statement:
   - If you specify BACKUP=YES on the PCCU definition statement, the resource cannot be used until a VARY ACQ command is issued for the NCP.
   - If you specify BACKUP=NO or use it by default on the PCCU definition statement:
     - The OWNER operand on the GROUP definition statement will be processed, but the corresponding line and physical unit cannot be used by this host.
     - A PU, and consequently, its LUs, with a different OWNER value than its higher-level resources cannot be activated.
4. The GROUP and LINE definition statements can contain more than one owner name, but the PU definition statement can specify only one OWNER value.
5. All PUs in the FRSESET should be defined with the same OWNER.
6. If a higher-level definition statement is not associated with this host, coding the OWNER operand on a lower-level definition statement does not establish ownership for this host.

PACING

```
►►─┬─PACING=(1,1,ADAP)──────────────────────────────────────────────────►◄
   │
   └─PACING=─┬─(1,1,ADAP)──────────────────────────────────────────────┬
             │                                                         │
             ├─max─────────────────────────────────────────────────────┤
             ├─(max──┬──────────────────────────────────────────────┬──)─┤
             │       ├─,,ADAP──────────────────────────────────────┤    │
             │       ├─,,FIXED─────────────────────────────────────┤    │
             │       └─,min─┬─┬─,ADAP─┬─┬──────────────────────────┤    │
             │              │ │       │ │                          │    │
             │              │ └─,FIXED┘ │                          │    │
             │              └───────────┘                          │    │
             │                                                     │    │
             └─(─┬─┬─,,ADAP─┬─┬─)──────────────────────────────────┘
                 │ │        │ │
                 │ └─,,FIXED┘ │
                 └────────────┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER*
dependencies:	*none*
range:	*0–63*

Specifies how pacing is handled between a logical unit and the NCP to which it is connected.

The formats for PACING values are:
- PACING=(*max*,*min*,*type*)
- PACING=*max*
- PACING=(*max*,*min*)
- PACING=(*max*,,*type*)
- PACING=(,,*type*)

max
Specifies the maximum window size. *max* is the maximum number of normal-flow requests that VTAM sends for a given LU-LU session before waiting for a pacing response. No further normal-flow requests can be sent to the logical unit until it is ready to receive more requests.

max can be expressed with leading zeros and is limited to eight digits.

min
Specifies the minimum window size.

min can be expressed with leading zeros and is limited to eight digits.

type
Specifies the pacing type: FIXED or ADAPTIVE (ADAP).

type is used only by NCP, and it requires at least NCP V5R4.

Defaults for PACING follow:
- If none of the values are specified, the defaults are PACING=(1,1,ADAP).
- If only *max* is specified, its value is also used for *min*. *type* defaults to ADAP.
- If *max* and *min* are specified, and *max* equals 0, then the defaults are PACING=(1,1,ADAP).
- If *max* and *min* are specified, and *max* does not equal 0, *max* must be greater than or equal to *min* for those values to be used. *type* is set to ADAP. If *max* is less than *min*, the values are set to PACING=(1,1,ADAP).
- If *max* and *type* are specified, *min* takes the value of *max*.
- If only *type* is specified, *max* and *min* are set to 1.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

PDELAY

```
   ┌─PDELAY=TERRESTR──────┐
   │                (1)   │
   ├─PDELAY=NEGLIGIB──────┤
►►─┼──────────────────────┼──────────────────────────────────────────────►◄
   │          ┌─TERRESTR─┐│
   └─PDELAY=──┼──────────┼┘
              ├─LONG─────┤
              ├─NEGLIGIB─┤
              └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PHYSRSC

```
►►─┬──────────────────────────────────────────────────────────────────────►◄
   └─PHYSRSC=──┬─pu_name──────────────┬─┘
               ├─NONE─────────────────┤
               └─(──┬─NONE──────┬──)─┘
                    ├─,──TIC3───┤
                    └─,──ODLC───┘
```

statements:	*GROUP*
dependencies:	*none*

Indicates the lines in the group are logical resources having dependency on a higher level (physical) resource, where *puname* specifies the label of a PU statement for the physical resource with which the logical lines are associated. A *puname* of NONE, or NONE as the first item in a list, indicates no hierarchical dependency between the physical and the logical resources.

Note: The PHYSRSC operand is processed only when a VTAM activates an NCP and becomes the owner of the NCP's resources. In a configuration where either of two VTAMs can activate the same NCP's resources (twin-tail

configuration), there will be a discrepancy between a display of resources after the NCP's resources are inactivated by the owning VTAM and later activated by the other VTAM.

For example, if the owning VTAM inactivates a frame relay physical line, and the other VTAM then activates the line, a display of a PU on the physical line shows the PU but does not show the logical lines which were displayed before the physical line was inactivated. The logical lines remained active on the owning VTAM. The PU is now active under the other VTAM but does not have any logical lines associated with it. To reassign the logical lines to the PU on the other VTAM, you would have to inactivate the NCP and reactivate it under the other VTAM, which would then be the owning VTAM.

POLIMIT

```
►►─┬─POLIMIT=(1,NOWAIT)──────────────────────────┬──────────────────────────────►◄
   │                                             │
   └─POLIMIT=(─┬─1─┬─────┬─,NOWAIT──────────┬─)──┘
               └─n─┘     │                  │
                         └─,─┬─NOWAIT──┬────┘
                             ├─QUEUE───┤
                             └─WAIT────┘
```

statements: *GROUP, LINE*
dependencies: *VTAM and NCP*

Must be coded for a polled nonswitched line. It determines the action taken by the NCP if the number (*n*) of negative responses is exceeded when polling.

Because the QUEUE parameter is required by NCP, you must code POLIMIT=(,QUEUE) or POLIMIT=(n,QUEUE).

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the POLIMIT operand.

PU

```
►►─┬─PU=YES──────┬──────────────────────────────────────────────────────────►◄
   │             │
   └─PU=─┬─YES─┬─┘
         └─NO──┘
```

statements: *GROUP, LINE*
dependencies: *VTAM only*

Specifies whether a BSC 3270 is treated as a physical unit and whether all terminals subordinate to the cluster are treated as logical units.

PU=NO
If PU=NO, it is assumed that the associated terminals are supported by a line control program such as the IBM licensed program Network Terminal Option (NTO) or a similar user-written program.

PU=YES
If PU=YES, the device type must be a 3270 type.

You cannot code this operand on the CLUSTER definition statement. You must code it on the GROUP or LINE definition statement, and let it sift down to the CLUSTER level.

PUDR

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─PUDR=──┬─NO──┬──┘
             └─YES─┘
```

statements: *GROUP, LINE, PU*
dependencies: *VTAM and NCP*

Specifies whether a PU is eligible for dynamic reconfiguration.

If you use sifting for PUDR and use VARY ACT,UPDATE=ALL to move a PU, make sure the sifted value is the same before and after any moves. If PUDR gets sifted differently as the result of a move and its value is no longer the same, VTAM and NCP will have mismatching PUDR values on the next initial activation of the NCP.

PUNAME

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─PUNAME=pu_name───┘
```

statements: *BUILD*
dependencies: *none*

Specifies the physical unit name that is to be associated with the NCP after activation is complete. The value coded here must match the name coded on the ID operand for the VARY ACT command issued to activate the NCP, and it must be a unique NCP name across the network. After the NCP is active, this physical unit name is the name used to identify the NCP (not the load module name specified on NEWNAME).

If you do not code PUNAME, the value of the NEWNAME operand will be used the physical unit name.

PUTYPE

```
    ┌─PUTYPE=5──────────┐
►►──┼───────────────────┼──────────────────────────────────────────►◄
    │                   │
    └─PUTYPE=──┬─5──┬──┘
               ├─1──┤
               ├─2──┤
               └─4──┘
```

statements: *GROUP, LINE, PU*

dependencies: *VTAM and NCP*

Specifies the physical unit type represented by the PU definition statement.

Notes:

1. If you code LNCTL=CA on the GROUP definition statement, 1, 2, and 5 are the valid values for PUTYPE, and PUTYPE=5 is the default. The physical unit with PUTYPE=1 is valid only for those groups created specifically for physical serial optical channel (SOC) lines. The physical unit with PUTYPE=2 is a channel adapter used to communicate with a type 2.1 node. The physical unit with PUTYPE=5 is a host processor or VTAM node.
2. If you code PUTYPE=4, DIAL=YES, and LNCTL=SDLC on the GROUP definition statement, the group is used for switched subarea connections.
3. For a type 2.1 node, code PUTYPE=2 and XID=YES.
4. For a non-switched frame relay PU (FRELAY=PHYSICAL), code PUTYPE=1 and XID=NO.
5. If you code PUTYPE=5, and if LNCTL=CA, you must specify ISTATUS=INACTIVE on the LINE definition statement. If you specify ISTATUS=ACTIVE, the activation fails.

For more details, see the *NCP, SSP, and EP Resource Definition Reference*

REGISTER

For Independent LUs:

```
►►─┬─REGISTER=NO──────────────────┬──────────────────────────────►◄
   │                              │
   └─REGISTER=─┬─NO──────────────┬┘
               ├─CDSERVR─────────┤
               └─NETSRVR─────────┘
```

statements: PU, LU
dependency: APPN only

For Dependent LUs:

```
►►─┬─REGISTER=NETSRVR─────────────┬──────────────────────────────►◄
   │                              │
   └─REGISTER=─┬─NETSRVR─────────┬┘
               ├─CDSERVR─────────┤
               └─NO──────────────┘
```

statements: PU, LU
dependency: APPN only

Specifies how a resource should be registered. dependent or independent LUs.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

REGISTER=NO
Indicates that the resource should not be registered.

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

RESETLU

```
►►─┬─RESETLU=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─RESETLU=─┬─YES─┬─────┘
              └─NO──┘
```

statements:	*PU, LU*
dependencies:	*dependent LUs only*

Specifies whether a DACTLU request followed by a ACTLU request will be sent during clean-up processing.

RESETLU=YES
Specifies that during clean-up processing for dependent LUs, VTAM will issue an ACTLU request after the DACTLU request.

RESETLU=NO
Specifies that during clean-up processing for LU-LU sessions, RESETLU=NO prevents a DACTLU-ACTLU from flowing to an LU.

RESSCB

```
►►─┬─RESSCB=0──────────────────────────────┬──────────────────────────►◄
   │                                       │
   └─RESSCB=number_of_session_control_blocks─┘
```

statements:	*LU*
dependencies:	*VTAM and NCP*
range:	*0–65535*

Specifies the number of session control blocks reserved by the NCP for the logical unit. This operand is valid for independent logical units only. If you code it for other logical units, VTAM ignores it.

For information about NCP restrictions on this operand, refer to the *NCP, SSP, and EP Resource Definition Reference*.

RETRIES

```
►►──────────────────────────────────────────────────────────────────►◄
   └─RETRIES=(,pause_interval,maximum_retries,)─┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*VTAM and NCP*
ranges:	*0–255* *0–127*

Specifies the number of attempts to be made to recover from errors occurring during transmission over the link. Each attempt at recovery, or retry, is a retransmission of data or control commands. Retries are performed in sets, which are called retry sequences.

pause_interval
Specifies the pause interval in seconds between two retry sequences. Valid values are 0–255. When the number of retries in a retry sequence is 0, *pause_interval* must be 0.

If *maximum_retries* is specified and valid and *pause_interval* is not specified, the NCP default of 0 is used for *pause_interval*.

maximum_retries
Specifies the maximum number of retry sequences. Valid values are 0–127, except for LNCTL=BSC or LNCTL=SS where valid values are 0–255.

If *pause_interval* is specified and valid and *maximum_retries* is not specified, the NCP default of 1 is used for *maximum_retries*.

If neither *pause_interval* nor *maximun_retries* is specified, the RETRIES operand is ignored. If a value that is not valid is specified for *pause_interval* or *maximum_retries*, an error message is given and the operand is ignored.

You can change the time interval and number of retry sequences of RETRIES for SDLC nonswitched PUs by dynamically adding the PUs using the DR ADD definition statement.

RNAME

```
►►──┬──────────────────────────────────────────────────────────────────────►◄
    └─RNAME=──┬─link_station_name──────────────┬─
              └─(─link_station_name──────────)─┘
```

statements:	*PCCU*
dependencies:	*none*

Must match the label on the PU definition statement that describes this NCP in an adjacent communication controller or host. (That PU definition statement identifies, to the other NCP or host, the link that connects it to this communication controller.) You can specify up to 13 SDLC or channel link station names; each one represents a contact point that a host can use to reach this communication controller. As you specify values for RNAME, keep in mind that VTAM reads multiple RNAMEs from right to left.

If you do not code RNAME when defining a link-attached communication controller, the VTAM operator should supply it as an operand on the VARY ACT command.

When the VTAM operator issues a VARY ACT command to activate this communication controller, if no link-station or channel-device address contact points are supplied (either with the RNAME and CUADDR operands or the VARY ACT command), a link station adjacent to the communication controller must already be active.

For information on how to code RNAME when the NCP is defined in a channel-attachment major node, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

SALIMIT

```
   ┌─SALIMIT=255──────────┐
►►─┼──────────────────────┼──────────────────────────────────────────────►◄
   │                      │
   │              ┌─255───┐│
   └─SALIMIT=─────┼───────┼┘
                  ├─511───┤
                  ├─1023──┤
                  ├─2047──┤
                  ├─4095──┤
                  ├─8191──┤
                  ├─16383─┤
                  ├─32767─┤
                  └─65535─┘
```

statements:	*BUILD, NETWORK*
dependencies:	*none*

Specifies the largest subarea address supported by this NCP in the native network.

If you code a value other than one of these, it is rounded up to the next valid value. If you code a value above 65535, 65535 is used.

SAVEMOD

```
   ┌─SAVEMOD=NO───────────┐
►►─┼──────────────────────┼──────────────────────────────────────────────►◄
   │              ┌─NO────┐│
   └─SAVEMOD=─────┼───────┼┘
                  └─YES───┘
```

statements:	*PCCU*
dependencies:	*3720, 3745 only*

Indicates whether you want a copy of the load module saved on the 3720 or 3745 communication controller external disk.

SAVEMOD=NO
If you code NO, the NCP is not saved.

SAVEMOD=YES
If you code YES, the NCP load module is saved after it is loaded from the host.

Notes:

1. This operand is used only if there is no SAVEMOD operand specified on the VARY ACT command, and if a load is actually performed.
2. You should not specify both DUMPLD=YES and SAVEMOD=NO on this PCCU definition statement. If you do, DUMPLD is forced to NO unless you specify LOADFROM=EXT on the VARY ACT command.

For more information on the VARY ACT command, refer to *OS/390 SecureWay Communications Server: SNA Operation*.

SECNET

```
   ┌─SECNET=NO──────┐
►►─┤                ├──────────────────────────────────────────►◄
   │          ┌─NO─┐│
   └─SECNET=──┤    ├┘
              └─YES┘
```

statements: *LINE, PU*
dependencies: *VTAM only*

Specifies whether a physical unit attached to a nonswitched line is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

This operand applies only if you have IBM 3710 Network Controllers or IBM 3174 token-ring attachments in your network.

The default for this operand is the value specified for SECNET on the LINE definition statement, if coded. However, if you do not code SECNET on the LINE definition statement, its default value on the PU definition statement is NO.

SECNET=NO
Indicates that data will not require special problem determination considerations.

SECNET=YES
Specifies data that is received by VTAM from this physical unit is flagged as requiring special problem determination considerations when routed to a communication network management (CNM) application.

SECURITY

```
   ┌─SECURITY=UNSECURE──────┐
►►─┤                        ├──────────────────────────────────────────►◄
   │              ┌─UNSECURE─┐│
   └─SECURITY=────┤          ├┘
                  ├─PUBLIC───┤
                  ├─UNDERGRO─┤
                  ├─SECURE───┤
                  ├─GUARDED──┤
                  ├─ENCRYPT──┤
                  └─SHIELDED─┘
```

statements: *GROUP, LINE, PU*
dependencies: *APPN only*

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SEGMENT

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └SEGMENT=──┬─1─┬──┘
               └─2─┘
```

statements: *PU*
dependencies: *VTAM and NCP*

Specifies which link segment a station is on and allows the NCP to set up Link Problem Determination Aid-2 (LPDA2) tests accordingly. SEGMENT is valid only if you specify LPDATS=LPDA2 on the LINE definition statement.

SEGMENT=1
Indicates the station is on segment number 1. This is the default value if TAILING=NO is specified.

SEGMENT=2
Indicates the station is on segment number 2. This is the default value if TAILING=YES is specified.

For more information on the TAILING operand, see the *NCP, SSP, and EP Resource Definition Reference*.

Note: VTAM uses this operand only when processing DR ADD definition statements for PUs that support LPDA-2.

Reference the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the SEGMENT operand.

SHM

```
    ┌─SHM=NO──────────┐
►►──┤                 ├──────────────────────────────────────────►◄
    │        ┌─NO─┐   │
    └SHM=────┤    ├───┘
             └YES─┘
```

statements: *GROUP*
dependencies: *VTAM only*

Specifies whether short-hold mode is supported. Only lines representing switched virtual circuits (SVCs) controlled by X.25 NCP Packet Switching Interface (NPSI) are capable of short-hold mode.

SHM=NO
Indicates that short-hold mode is not supported.

SHM=YES
Indicates that short-hold mode is supported and any short-hold mode operands specified on the PATH definition statement in the switched major node should be used for the connection. If you specify SHM=YES, code DIAL=YES and PUTYPE=4 on the GROUP definition statement.

SSCPFM

```
  ┌─SSCPFM=FSS──────────┐
►►┼─────────────────────┼──────────────────────────────────────────►◄
  │                     │
  │            ┌─FSS────┐
  └─SSCPFM=────┼────────┼
               ├─USSNTO─┤
               ├─USSSCS─┤
               ├─USS3270┤
               ├─USS3275┤
               └─USS3780┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*
dependencies: *VTAM only*

Determines whether a logical unit or terminal can support character-coded messages in its communication with the SSCP.

For BSC devices, the only valid values are USS3275 and USS3270. For non-SNA devices supported with Network Terminal Option (NTO), the only valid values are USS3780 and USSNTO.

For BSC devices, the default is USS3275. For all other devices, the default is SSCPFM=FSS.

The SSCPFM operand has no effect on BSC terminals that are not printers.

SSCPFM=FSS
Specifies that formatted messages are supported by this logical unit or terminal for SSCP communication. Consult the individual terminal component description manual to determine whether this value or SSCPFM=USSSCS should be coded.

SSCPFM=USSNTO
Coded for all non-SNA devices supported with NTO for which USS3780 does not apply. These devices use the USS command facilities and are supported by all releases of NTO. Refer to the *NTO Installation* for more information.

SSCPFM=USSSCS
Specifies that character-coded messages are supported by this logical unit or terminal for SSCP communication. Consult the individual terminal component description manual to determine whether this value or SSCPFM=FSS should be coded.

SSCPFM=USS3270
Allows printing of a user-defined logon message (USSMSG10) on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

SSCPFM=USS3275
Prevents printing of a user-defined logon message (USSMSG10) for a terminal defined as an SDLC 3275 (PU type 1) that might have an attached printer. Code SSCPFM=USS3275 to prevent user-defined logon messages from printing on BSC printers.

Note: Coding SSCPFM=USS3275 prevents the local copy hardware function key from working with certain BSC printers. To make sure the local copy function works, code SSCPFM=USS3270.

SSCPFM=USS3780
Coded for non-SNA devices, supported with the network terminal option (NTO), that use the 3780 protocol. These devices use USS command facilities (LOGON and IBMTEST) and are supported only by Release 2 of NTO.

SUBAREA

```
►►─┬─SUBAREA=1──────────────┬──────────────────────►◄
   └─SUBAREA=─┬─1──────────┬─┘
              └─suba_addr──┘
```

```
►►─┬──────────────────────────┬──────────────────►◄
   └─SUBAREA=suba_num─────────┘
```

statements:	HOST	*statements:*	PCCU
dependency:	none	*dependency:*	none

```
►►─SUBAREA=suba_addr──────────────────►◄
```

statements:	BUILD, NETWORK
dependency:	none

On PCCU: Specifies the subarea number of the host to which this PCCU definition statement applies. *suba_num* is a decimal integer in the range 1–65535.

The subarea value should match the value assigned to the HOSTSA start option when VTAM is started. VTAM uses the subarea value to determine which PCCU definition statement it should process in the NCP major node. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for a discussion of how VTAM selects a particular PCCU definition statement.

On BUILD: Specifies the native network subarea address that represents the gateway NCP to the network being defined. This subarea address is unique only in the network being defined by this BUILD definition statement.

This operand is required.

On NETWORK: Specifies the subarea address that represents the gateway NCP to the network being defined. This subarea is unique only in the network being defined by this NETWORK definition statement.

On HOST: Specifies the subarea address of the access method or access methods that the NCP can communicate with over a channel.

Each subarea in a network (whether the network is controlled by one or multiple NCPs) must have a unique subarea address. In SNA interconnected networks, this is true for each network that the gateway NCP joins.

SUBPORTS

```
►►──SUBPORTS=──┬─(puname1,puname2)──────────────────┬──────────────────────────►◄
               ├─(puname1,puname2,puname3,)─────────┤
               ├─(puname1,puname2,puname3,puname4)──┤
               ├─(puname1,puname2,,puname4)─────────┤
               └─(puname1,puname2,,)────────────────┘
```

statements: *FRSESET*
dependencies: *PU type 1 only*

Specifies the names of the primary type 1 PUs and their optional backups. There are a minimum of two and a maximum of four PUs in SUBPORTS.

SWNORDER

```
      ┌─SWNORDER=value_of_swnorder_start_option──────┐
►►────┤                                              ├──────────────────────────►◄
      │                ┌─value_of_swnorder_start_option─┐
      └─SWNORDER=──────┤                                ├──┘
                       ├─CPNAME───────────────────────┤
                       └─STATNID──────────────────────┘
```

statements: *GROUP, LINE*
dependencies: *none*

Specifies the way VTAM locates a switched PU. If you do not code SWNORDER, the default is the value of the SWNORDER start option.

If VTAM does not find the PU using the SWNORDER value you specify, it searches by the station identifier. For example, if you code SWNORDER=CPNAME and VTAM cannot find a PU with a matching CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

This operand overrides the SWNORDER start option value.

SWNORDER=CPNAME
Specifies that VTAM searches for a switched PU by the CPNAME first.

SWNORDER=STATNID
Specifies that VTAM searches for a switched PU by the station identifier first.

TERM

```
►►──┬──────────┬──────────────────────────────►◄   ►►──TERM=type──────────────────────────►◄
    └TERM=type─┘
```

statements:	PU, LU	*statements:*	GROUP, LINE, CLUSTER, TERMINAL
dependency:	VTAM and NCP	*dependency:*	VTAM and NCP

Describes, for a virtual logical unit supported through Network Terminal Option (NTO), the device data stream compatible characteristics. These characteristics are placed in an 8-byte program-supplied storage area, after an INQUIRE DEVCHAR macroinstruction is executed by an application program that specifies the virtual logical unit name. For more information on the DEVCHAR DSECT, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

For specific considerations for BSC terminals, refer to the *NCP, SSP, and EP Resource Definition Reference*.

Refer to the *Network Terminal Option Installation* for more information.

TGN

For subarea TGN:

```
►►──┬─TGN=1──────────────────┬──────────────────────────►◄
    │                        │
    └TGN=──┬─1──────────────┬─┘
           │                │
           ├─ANY────────────┤
           │                │
           └─tg_number──────┘
```

For APPN TGN:

```
►►──┬─TGN=ANY────────────────┬──────────────────────────►◄
    │                        │
    └TGN=──┬─ANY────────────┬─┘
           │                │
           └─tg_number──────┘
```

statements:	PCCU, PU	*statements:*	PCCU, PU
dependency:	VTAM and NCP	*dependency:*	*VTAM and NCP; CPNAME; NETID*
range:	1–255	*range:*	0–20

Subarea: Assigns the transmission group (TG) number that represents the connection between two nodes. A connection can be:

- The channel between host and communication controller, where PUTYPE=5 is coded
- The channel between two communication controllers, where PUTYPE=4 is coded.

APPN: Specifies the transmission group (TG) number that represents the connection between this node and the destination APPN node. For APPN, this operand is valid when PUTYPE=2.

By specifying a *tg_number*, you indicate a preferred TG that might be used depending on how the partner nodes negotiate the number.

For APPN-capable nodes, the CPNAME and NETID operands are required with the TGN operand. Note that you can specify the CPNAME operand without the TGN

operand and the transmission group number will be negotiated. However, if a value for TGN is specified, then a value for CPNAME and NETID must also be specified.

TGN=*tg_number*
Specifies a decimal number assigned to the transmission group.

For APPN, coding TGN=0 means that the PU does not support parallel TGs.

TGN=ANY
For subarea, allows the adjacent node to accept whatever valid transmission group number it receives on the XID. You can specify TGN=ANY for only one end of the transmission group. For the other end, specify a valid TGN number or let it default to 1. If you specify TGN=ANY for both ends, an error occurs. For APPN, coding TGN=ANY, or not coding TGN, allows the transmission group number to be negotiated. For connections to LEN nodes, or to any node that does not support parallel TGs, TGN=0 is always assumed.

Note: VTAM does not validate TGN when coded for subarea resources.

TGP

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └TGP=tgp_name──┘
```

statements: *GROUP, LINE, PU*
dependencies: *APPN only*

Specifies the name of a transmission group (TG) profile definition. The characteristics of the TG profile (along with any modifiers) become the characteristics of the PU. If TGP is not specified or has not been activated when the PU becomes active, default TG characteristics are assigned.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TOPO

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └TOPO=CONNECT──┘
```

statements: *PU*
dependencies: *APPN only*

Controls the way APPN connections using this link station will be reported to APPN topology and routing services for inclusion into APPN functions.

Note: Except in the switched major node, the only value for TOPO is CONNECT. If you specify APPN on the CONNTYPE start option, it is not necessary to code TOPO, as you automatically get a value of CONNECT.

TOPO=CONNECT
APPN topology and routing services receives information about the connection

and its characteristics when the connection is activated. When the connection is deactivated, APPN topology and routing services receives notification that the transmission group is inactive.

UPARM1

```
►►─┬─UPARM1=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM1=user-defined_operand_value──┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
►►─┬─UPARM2=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM2=user-defined_operand_value──┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM3=user-defined_operand_value──┘
```

statements:	*GROUP, LINE, PU*
dependencies:	*APPN only*

Specifies a user-defined operand value.

USE

```
┌─USE=NCP──────────┐
►►──┤                ├──────────────────────────────────────────►◄
└─USE=─┬─NCP──┬─┘
       ├─EP───┤
       ├─REDEF┤
       └─SPARE┘
```

statements: *GROUP, LINE*
Note: code USE=REDEF, USE=SPARE only on LINE.

dependencies: *VTAM and NCP*

Specifies the initial operating condition of the line for lines that can operate alternately in network control mode and emulation mode.

USE also allows you to specify SDLC lines in the NCP generation for line slots that do not have physical lines attached, but which can be attached later, and allows you to redefine a line that is attached.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the USE operand.

USE=EP
Indicates that the line operates initially in emulation mode. Prevents the line from being activated unless a VARY ACT command is issued for it. Activating such a line with a VARY command gives the line to VTAM and NCP, whether or not it is in use by EP.

USE=NCP
Specifies that the line operates initially in network control mode.

USE=REDEF
Specifies that the line is being used, but can be changed to a spare line. The mode of operation is assumed to be NCP. USE=REDEF is used with the MODIFY command to change it from a defined line to a spare line and vice versa. Note that a line defined with USE=REDEF must be the only line in a line group.

USE=SPARE
Specifies that the line is being defined for later use, although there is no line attached to that line slot in the NCP.

The mode of operation is assumed to be NCP. Note that a line defined with USE=SPARE must be the only line in a line group.

USSTAB

```
┌─USSTAB=ISTINCDT──────────┐
►►──┤                          ├──────────────────────────────────────────►◄
└─USSTAB=uss_table_name─┘
```

statements: *GROUP, LINE, PU, LU, CLUSTER, TERMINAL*

dependencies: *VTAM only; dependent LUs only*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit. USSTAB is valid for dependent logical units only.

A terminal user can issue a USS command with the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

If you code USSTAB on a CLUSTER or TERMINAL definition statement, you must also code PU=YES on the GROUP or LINE definition statement.

VERALSID

```
                ┌─VERALSID=NO──────────┐
►►──┬──────────┼──────────────────────┼──────────────────────────────────────────────►◄
    │          │                      │
    │          │              ┌─NO─┐  │
    │          └─VERALSID=────┼────┼──┘
    │                         └─YES─┘
    │          ┌─VERALSID=NO─┐   (1)
    │          │             │
    │          └─VERALSID=NO─┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

statements: *GROUP, LINE, PU*
dependencies: *APPN only*

Specifies whether the adjacent link station name should be used to determine the link station.

VERALSID=NO
Specifies that the adjacent link station name is not used to determine the link station.

VERALSID=YES
Specifies that the adjacent link station name is used to determine the link station. When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to XID3s. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

VFYC

```
►►─┬─VFYC=YES──────────┬──────────────────────────────────────────►◄
   │                   │
   └─VFYC=─┬─YES───────┬─┘
           ├─IGNORE────┤
           └─NO────────┘
```

statements: *PCCU*
dependencies: *none*

Specifies what action VTAM should take if the correlators on the NCP load module and the NCP/EP definition facility (NDF) resource resolution table (RRT) do not match. When the correlators do not match, VTAM can take one of the following actions:

VFYC=IGNORE
Specifies that VTAM continues the current activation process.

VFYC=NO
Specifies that VTAM reloads the communication controller.

Note: If VFYC=NO, an interruption of all the NCP-attached hosts occurs. The NCP load module and the RRT in this host must match. If they do not match, a repeated reload of the NCP occurs until an operator deactivates it.

VFYC=YES
Specifies that VTAM asks the operator whether to reload the communication controller with the requested NCP, stop the activation process, or ignore the mismatch.

VFYC is checked only after a VFYLM check reveals a match between the load module name and the subarea.

VFYLM

```
►►─┬─VFYLM=NO──────────┬──────────────────────────────────────────►◄
   │                   │
   └─VFYLM=─┬─NO────────┬─┘
            └─YES───────┘
```

statements: *PCCU*
dependencies: *none*

Specifies how VTAM handles mismatches between the name and subarea of the load module currently in the NCP and the NEWNAME operand on the BUILD definition statement. When the names or subareas do not match, one of the following actions occurs:

VFYLM=NO
Specifies that the operator must load the NCP with the different load module.

VFYLM=YES
Specifies that VTAM asks the operator whether to reload the communication controller with the requested NCP or stop the activation process.

When the load module name matches VTAM's major node name, the value of AUTOSYN applies.

VNGROUP

```
                          (1)
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └VNGROUP=group_name────┘
```

Notes:

1 Code VNGROUP if you code VNNAME.

statements: *GROUP, LINE*
dependencies: *VTAM only; VNNAME; ECLTYPE=PHYSICAL on GROUP*

Specifies the name of the logical GROUP containing dial-out links available for use on the connection network named on the VNNAME operand.

For sessions that are routed through the connection network, a dynamic PU is created to represent the destination node. In effect, DYNPU=YES is enforced automatically when a session is established through the connection network. The name specified on the VNGROUP operand provides the name of the group for the dynamic PU that is allocated.

VNNAME

```
                                   (1)
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └VNNAME=connection_network_cpname┘
```

Notes:

1 Code VNNAME if you code VNGROUP.

statements: *GROUP, LINE*
dependencies: *VTAM only; VNGROUP; ECLTYPE=PHYSICAL on GROUP*

Specifies a 1–17 character network-qualified CPNAME for the connection network. VNNAME is reported to the network topology as a virtual node and is treated as an adjacent CP to this node. If the name is not network-qualified, the NETID of the SSCP that owns the connection network (NETID of the host) is used.

For sessions that are routed through the connection network, a dynamic PU is created to represent the destination node. In effect, DYNPU=YES is enforced automatically when a session is established through the connection network.

VPACING

```
   ┌─VPACING=(2,1,ADAP)──────────────────────┐
►►─┼─────────────────────────────────────────┼──────────────────────────────►◄
   │                ┌─(2,1,ADAP)─────────────┐│
   └─VPACING=───────┼────────────────────────┼┘
                    │─max────────────────────│
                    │─(max──┬──────────────┬─)─│
                    │       │─,,ADAP───────│  │
                    │       │─,,FIXED──────│  │
                    │       │      ┌─,ADAP─┐│  │
                    │       └─,min─┼───────┼┘  │
                    │              └─,FIXED─┘   │
                    │   ┌─,,ADAP─┐              │
                    └─(─┼────────┼─)────────────┘
                        └─,,FIXED─┘
```

statements:	*GROUP, LINE, PU, LU, CLUSTER*
dependencies:	*VTAM only*
range:	*0–63*

Specifies how pacing is handled between VTAM and the NCP.

The formats for VPACING values are:
- VPACING=(*max*,*min*,*type*)
- VPACING=*max*
- VPACING=(*max*,*min*)
- VPACING=(*max*,,*type*)
- VPACING=(,,*type*)

max
Specifies the maximum window size. *max* is the maximum number of normal-flow requests that VTAM sends for a given LU-LU session before waiting for a pacing response. No further normal-flow requests can be sent to the logical unit until it is ready to receive more requests.

max can be expressed with leading zeros and is limited to eight digits.

min
Specifies the minimum window size.

min can be expressed with leading zeros and is limited to eight digits.

type
Specifies the pacing type: FIXED or ADAPTIVE (ADAP).

type is used only by NCP, and it requires at least NCP V5R4.

Defaults for VPACING follow:
- If none of the values are specified, the defaults are VPACING=(2,1,ADAP).
- If only *max* is specified, its value is also used for *min*. *type* defaults to ADAP.
- If *max* and *min* are specified, and *max* equals 0, then the defaults are VPACING=(2,1,ADAP).
- If *max* and *min* are specified, and *max* does not equal 0, *max* must be greater than or equal to *min* for those values to be used. *type* is set to ADAP. If *max* is less than *min*, the values are set to VPACING=(2,1,ADAP).

- If *max* and *type* are specified, *min* takes the value of *max*.
- If only *type* is specified, *max* and *min* are set to 1.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

VTAMTOPO

```
►►─┬─VTAMTOPO=REPORT──────────┬──────────────────────────────────────►◄
   │                          │
   └─VTAMTOPO=─┬─REPORT───┬───┘
               │          │
               ├─NOREPORT─┤
               ├─NOLLINES─┤
               ├─NOSWPUS──┤
               ├─INCLUDE──┤
               └─IGNORE───┘
```

statements: *BUILD*
dependencies: *none*

```
►►─┬──────────────────────────┬──────────────────────────────────────►◄
   │                          │
   └─VTAMTOPO=─┬─REPORT───┬───┘
               │          │
               ├─NOREPORT─┤
               ├─NOLLINES─┤
               └─NOSWPUS──┘
```

statements: *GROUP*
dependencies: *(switched logical line)*

```
►►─┬──────────────────────────┬──────────────────────────────────────►◄
   │                          │
   └─VTAMTOPO=─┬─REPORT───┬───┘
               │          │
               └─NOREPORT─┘
```

```
►►──────────────────────────────────────────────────────────────────►◄
```

statements: *LINE*
dependencies: *(switched logical line)*

Specifies whether the VTAM topology agent reports the major nodes or subordinate minor nodes to the CMIP manager application programs for SNA local topology.

Specifying VTM.VTAMTOPO allows VTAMTOPO to be handled appropriately by the NCP Definition Facility for versions of NDF that do not recognize VTAMTOPO natively.

Note: Refer to “Appendix C. VTAMTOPO Filtering Option Reporting” on page 837 for more information on VTAMTOPO filtering option reporting.

Table 32. NCP Line Definition Dependency for Coding VTAMTOPO Keyword

Logical Line Type	Dependency
Token Ring	DIAL=YES and ECLTYPE=LOGICAL
Frame Relay	DIAL=YES and FRELAY=LOGICAL or LOGICAL, PERIPHERAL
X25 (NPSI, ODLC)	PHYSRSC=puname and DIAL=YES
ESCON	DIAL=YES and PHYSRSC=puname

VTAMTOPO=REPORT

On BUILD: Specifies that the VTAM topology agent reports this NCP major node and all subordinate minor nodes, including connected switched PUs under switched logical lines. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent reports all subordinate lines and connected switched PUs. VTAMTOPO is valid only for a switched logical line group, see Table 32. This does not override any value specified on the LINE.

On LINE: Specifies that the VTAM topology agent reports this logical LINE. VTAMTOPO is valid only for a switched logical line, see Table 32.

VTAMTOPO=NOREPORT

On BUILD: Specifies that the VTAM topology agent does not report the switched logical lines or connected switched PUs. The topology agent still reports the major node and the other subordinate resources. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent does not report any subordinate lines and connected switched PUs. VTAMTOPO is valid only for a switched logical line group. This does not override any value specified on the LINE.

On LINE: Specifies that the VTAM topology agent does not report this logical LINE. VTAMTOPO is valid only for a switched logical line, see Table 32.

VTAMTOPO=NOLLINES

On BUILD: Specifies that the VTAM topology agent does not report the switched logical lines. The topology agent still reports the major node and the other subordinate resources, including the connected switched PUs. This does not override any value specified on the GROUP or LINE.

On GROUP: Specifies that the VTAM topology agent does not report any subordinate switched logical lines. The topology agent still reports the connected switched PUs under the subordinate switched logical lines. VTAMTOPO is valid only for a switched logical line group. This does not override any value specified on the LINE.

VTAMTOPO=NOSWPUS

On BUILD: Specifies that the VTAM topology agent does not report the connected switched PUs. The topology agent still reports the major node and the other subordinate resources, including the switched logical lines. This does not override any value specified on the GROUP.

On GROUP: Specifies that the VTAM topology agent does not report any connected switched PUs under the subordinate switched logical lines. VTAMTOPO is valid only for a switched logical line group. The topology agent still reports the subordinate switched logical lines.

VTAMTOPO=INCLUDE
On BUILD: Specifies that the VTAM topology agent reports this NCP major node and all subordinate minor nodes. VTAMTOPO=INCLUDE overrides any other value specified on the GROUP or LINE.

VTAMTOPO=IGNORE
On BUILD: Specifies that the VTAM topology agent does not report this NCP major node nor any subordinate minor nodes. VTAMTOPO=IGNORE overrides any other value specified on the GROUP or LINE.

XID

```
►►─┬─XID=NO──────────┬──────────────────────────────────────────►◄
   │                 │
   └─XID=─┬─NO──┬────┘
          └─YES─┘
```

statements: *GROUP, LINE, PU*
dependencies: *VTAM and NCP*

Specifies whether a physical unit on a nonswitched SDLC line can identify itself to VTAM through XID.

XID=NO
Indicates an XID is not sent to the physical unit.

For non-switched frame relay PUs, code XID=NO.

XID=YES
Indicates an XID is sent to the physical unit. Code for all physical units that have the ability to receive and respond to an XID in normal disconnect mode.

XID=YES is required for type 2.1 peripheral nodes.

XNETALS

```
►►─┬─XNETALS=value_of_XNETALS_start_option─┬──────────────────────────►◄
   │                                       │
   └─XNETALS=─┬─NO──┬──────────────────────┘
              └─YES─┘
```

statements: *PU*
dependencies: *none*

determines whether the network ID of an attaching node or VTAM's network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true. If the NETID that the PU sent in its XID is not the same as VTAM's and XNETALS=NO or is defaulted, IST680I will be issued.

XNETALS=NO

Indicates that VTAM's network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement, then it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement, then VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES

Indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Note: Providing that XNETALS=YES is coded, you can prohibit nonnative network attachment for LU-LU sessions for a particular PU by coding XNETALS=NO on the GROUP, LINE or PU statements.

When using XNETALS=NO, VTAM indicates to the NCP that this PU will use VTAM's native network ID.

Switched Major Node

To define a switched major node containing definitions for switched connections to subarea or peripheral nodes, code a switched line within a local switched line group, and code a corresponding remote physical unit in the switched major node.

The switched lines are defined within switched line groups within the following major nodes:
- External communication adapter (XCA)
- NCP

These switched connections can be made to a communication controller that is attached to a host processor. These connections can also be made to a host through an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter, or through an IP network for Enterprise Extender.

The remote physical unit can be a type 1, 2, 4, or 5 physical unit that is defined within a switched major node.

PU and PATH Definition Statements for a Switched Subarea

The PU definition statements define subareas attached over switched lines that can be dialed into or dialed out from a communication controller using NCP.

PU and PATH Definition Statements for the XCA Major Node

The PU definition statement defines peripheral physical units on switched lines that can be dialed into or dialed out from the host through an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter, or through an IP network using Enterprise Extender. The lines are defined in switched line groups in the external communication adapter major node. For dial-out operations, the PATH definition statement defines the paths used to connect the physical unit to the host through an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter, or through an IP network using Enterprise Extender.

Table 33. Switched Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=SWNET	462	R			
		CONFGDS	416				
		CONFGPW	416				CONFGDS
		MAXDLUR	445				NN and ICN only; DLURNAME
		MAXGRP	445				PATH
		MAXNO	445				PATH
		VTAMTOPO	466			REPORT	
name	GROUP	VTAMTOPO	466				
name	PU	ADDR	411	R			nonswitched connections
		ANS	411		G	STOP	
		ASDP	412		G	NO	
		AUTHLEN	413		G	YES	Type 2.1; APPN only
		CAPACITY	414		G	8K 25M (ATM native connections)	APPN only
		CONNTYPE	417		G	CONNTYPE start option	APPN only; NODETYPE

Table 33. Switched Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		COSTBYTE	418		G	0	APPN only
		COSTTIME	418		G	0	APPN only; TOPO
		CPCP	418		G	CPCP start option	APPN only
		CPNAME	419				Type 2.1
		DATMODE	419		G	HALF	
		DISCNT	422		G	(NO)	PUTYPE=2
		DWACT	430			NO	
		DYNADJCP	430		G	DYNADJCP start option	APPN only; CPNAME
		DYNLU	431		G	DYNLU start option	PUTYPE=2; CDRDYN
		ENCRTYPE	432		G,P	DES	
		ENCR	432		G,P	NONE	
		HPR	436		G		APPN only
		IDBLK	436				CPNAME; IDNUM
		IDNUM	437				CPNAME; IDBLK; PUTYPE=4,5
		INCLUD0E	438			NO	
		IRETRY	439		G	NO	PUTYPE=1,2
		ISTATUS	440		G	ACTIVE	
		LIMRES	440		G	NO	LIMQSINT; LU 6.2; PUTYPE=1, 2; switched dynamics
		LLERP	441			NOTPREF	Type 2.1; HPR=YES
		LUGROUP	443				
		LUSEED	443				LUGROUP
		MAXDATA	444		G	261 (PU type 1) 265 (PU type 2)	see description
		MAXOUT	446		G	1	3172 Controller
		MLTGPRI	447			0	
		NATIVE	448		G		BN=YES APPN only
		NETID	229				XNETALS
		NN	450				APPN only
		PASSLIM	451		G	1	PUTYPE=1,2
		PDELAY	451		G	TERRESTR	APPN only
		PRTCT	452				PUTYPE=4,5
		PUTYPE	452		G	2	
		SECNET	456			NO	PUTYPE=1,2
		SECURITY	456		G	UNSECURE	APPN only
		SPAN	458				NetView
		SUBAREA	459			*in_xid_value*	PUTYPE=4,5
		TGCONF	459			MULTI,NORMAL	PUTYPE=4, Not valid for X.25
		TGN	460				see description
		TGP	461		G		APPN only

Table 33. Switched Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		TOPO	461			CONNECT	APPN only
		UPARM1	462		G	128	APPN only
		UPARM2	463		G	128	APPN only
		UPARM3	463		G	128	APPN only
		VERALSID	464		G	NO	APPN only
		VTAMTOPO	466		G		
		XNETALS	468			XNETALS start option value	
name	PATH	CALL	414			INOUT	GRPNM
		DIALNO	420				LINENM
		DLCADDR	423				LINENM
		DLURNAME	429				DLCADDR
		GID	435				
		GRPNM	435				CALL VERID VERIFY
		HOSTNAME	436				
		IPADDR	439				
		LINENM	441				DIALNO DLCADDR
		PID	452				
		REDIAL	453			3	
		SAPADDR	455				
		SHM	457			NO	DIALNO GRPNM
		SHMTIM	457			0.0	
		USE	463			YES	
		VERID	465				GRPNM
		VERIFY	465			NONE	GRPNM; VERID
name	LU	LOCADDR	441	R	G		
		ASLENT	412		G,P		ASLTAB PUTYPE=1,2
		ASLTAB	412		G,P		PUTYPE=1,2
		BATCH	413		G,P	NO	pre-V4R3 NCPs PUTYPE=1,2
		CERTIFY	415		P	NO	ENCR
		CKEYNAME	415				
		CLRSESSQ	416				
		DLOGMOD	429		G,P		PUTYPE=1,2
		ENCR	432		G,P	NONE	
		ENCRTYPE	432		G,P	DES	
		FASTPASS	433		G,P	YES	TSO only
		FEATUR2	434		G,P	(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	
		ISTATUS	440		G,P	ACTIVE	
		LOGAPPL	442		G,P		PUTYPE=1,2

Table 33. Switched Major Node Definition Statements and Operands (continued)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		LOGTAB	443		G,P		PUTYPE=1,2
		MAXSESS	446		G,P		NCP-attached independent LUs
		MDLENT	446		G,P		MDLTAB PUTYPE=1,2
		MDLTAB	447		G,P		PUTYPE=1,2
		MODETAB	447		G,P	ISTINCLM	PUTYPE=1,2
		PACING	450		G,P	1	PUTYPE=1,2
		REGISTER	454		P	see description	APPN only
		RESETLU	454		P	YES	
		RESSCB	455			0	independent LU
		SPAN	458				NetView
		SSCPFM	458		G,P	FSS	PUTYPE=1,2
		TERM	459		G,P		PUTYPE=1,2
		USSTAB	463		G,P	ISTINCDT	PUTYPE=1,2; dependent LUs
		VPACING	466		G,P	2	PUTYPE=1,2

Legend:

R Required operand.

G Code this operand on the higher-level GROUP definition statement to take advantage of the sift effect, or code it on this definition statement.

P Code this operand on the higher-level PU definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

Enterprise Extender Connections

VBUILD Definition Statement

Code a VBUILD definition statement for each switched major node.

PU Definition Statement

Code a PU definition statement for each physical unit in the switched major node.

PATH Definition Statement

Code a PATH definition statement to define a path to a physical unit in a switched major node. You can code up to 256 PATH definition statements for each physical unit. The PATH definition statement must immediately follow the PU definition statement that defines the associated physical unit. VTAM searches the PATH definition statements for an available path in the order given in the configuration deck.

The IPADDR keyword specifies the remote IP address (in dotted decimal format) for a dial-out connection. If the remote APPN node is a CS for OS/390 node, the address must be the VIPA address the remote node is currently using for the

Enterprise Extender connections. For example, the VIPA address specified or defaulted to on the IPADDR VTAM start option or modified by the F VTAMOPTS, IPADDR operator command.

The HOSTNAME keyword specifies the name of the remote (destination) host name for a dial-out connection. IPADDR and HOSTNAME keywords are mutually exclusive. Use of this keyword implies that the local CS for OS/390 node can perform TCP/IP name-to-address resolution, either by connectivity to a DNS name server, or by using a local, HOSTS.SITEINFO dataset. If the host name is specified and cannot be resolved, the dial will fail. For more information refer to the *OS/390 SecureWay Communications Server: IP Configuration*. The DNS name servers and HOSTS.SITEINFO must be configured to only return VIPA addresses if the name represents a CS for OS/390 APPN node. Furthermore, the VIPA address returned must be the one specified or defaulted to on the VTAM IPADDR start option or modified by the F VTAMOPTS, IPADDR operator command. For information on displaying information on networks and hosts, refer to the *OS/390 SecureWay Communications Server: IP User's Guide*.

In order to use HOSTNAME, VTAM must be able to find one or more TCPIP definition files. This is done via the Native MVS Sockets resolver. Do not confuse this with the LE Resolver, sometimes referred to as the OS/390 Resolver. If you intend to use a name server to resolve HOSTNAME, VTAM must be able to find the TCPIP.DATA file, which in turn, contains the IP address(es) of nameserver(s) to use. It also contains other necessary information such as the local domain name, local host name, etc. If you intend to use a local HOST table instead of a name server, there also is a specific search order to find this dataset. Refer to the *OS/390 SecureWay Communications Server: IP Configuration*, in the ″Resolver Configuration Files″ section for information on the search order for Local Hosts tables.

Note: These datasets cannot be concatenated like many MVS datasets can.

The SAPADDR keyword specifies the remote service access point (RSAP) value.

Note: The GRPNM keyword ties the switched major node PATH statement to the XCA major node. The name on the XCA GROUP statement must match the value on the GRPNM in the SMN PATH.

LU Definition Statement

Code an LU definition statement for each logical unit associated with a type 1 or 2 physical unit within a switched major node. The LU definition statement must follow the PU definition statement that defines the physical unit with which the logical unit is associated.

Dynamic Change of Operands

You can dynamically change the following switched major node operands by editing the VTAMLST and then issuing the VARY ACT,UPDATE=ALL command. You can specify the new operand value on a higher-level definition statement for sifting, if applicable.

ADDR
ANS
ASDP
ASLENT
ASLTAB
BATCH
CALL
CAPACITY
CONNTYPE
COSTBYTE
COSTTIME
CPCP
DATMODE
DIALNO
DISCNT
DLCADDR
DLOGMOD
DLURNAME
DYNADJCP
DYNLU
ENCR
ENCRTYPE
FASTPASS
FEATUR2
GID
GRPNM
HOSTNAME
IDBLK
IDNUM
IPADDR
IRETRY
ISTATUS
LIMRES
LINENM
LOCADDR
LOGAPPL
LOGTAB
LUGROUP
LUSEED
MAXDATA
MAXOUT
MAXSESS
MDLENT
MDLTAB
MODETAB
NN
PACING
PASSLIM
PDELAY
PID
PUTYPE
REDIAL
REGISTER
SAPADDR
SECURITY
SECNET
SHM
SHMTIM
SSCPFM
TERM
TGP
TOPO
UPARM1
UPARM2
UPARM3
USE
USSTAB
VERALSID
VERID
VERIFY
VPACING

In addition to the above VTAM operands, you can also dynamically change the following NCP operands, which are passed to the NCP during switched PU connection. For information on these operands, refer to the *NCP, SSP, and EP Resource Definition Reference*.

- COMRATE
- DLCI
- DYNWIND
- HPRQLIM
- INTFACE
- IPQLIM
- LADDR
- SNETMASK
- T1TIMER
- T2TIMER

To dynamically change an operand on the PU or LU statement, the resource to which it applies must be inactive. For instance, an operand on an LU definition statement that receives its value from a PU definition statement can be dynamically changed only when the LU is inactive; the PU can remain active.

You can dynamically change the operands associated with a PATH statement or tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) when the PU is active, connectable, or inactive. You cannot change the PATH operands when the PU is in pending state.

Refer to *OS/390 SecureWay Communications Server: SNA Operation* and the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on the VARY ACT,UPDATE=ALL command.

Note: For a switched major node, you cannot add APPN capability or SDDLU support to a PU with the dynamic change function. If a PU has not been coded as APPN-capable, you cannot add the APPN operands to it. Likewise, if you have not coded LUGROUP (which allows SDDLU support) on a PU definition statement, you cannot add it with dynamic change.

NODETYPE Start Option

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

XCA and NCP Connections

VBUILD Definition Statement

Code a VBUILD definition statement for each switched major node.

PU Definition Statement

Code a PU definition statement for each physical unit in the switched major node. Operands specified on a PU definition statement override any specified under the NCP.

PATH Definition Statement

Code a PATH definition statement to define a path to a physical unit in a switched major node. You can code up to 256 PATH definition statements for each physical unit. The PATH definition statement must immediately follow the PU definition statement that defines the associated physical unit. VTAM searches the PATH definition statements for an available path in the order given in the configuration deck.

A PATH definition statement is required if data terminal equipment (DTE) ID verification or call selection is to be used on an incoming call.

LU Definition Statement

Code an LU definition statement for each logical unit associated with a type 1 or 2 physical unit within a switched major node. The LU definition statement must follow the PU definition statement that defines the physical unit with which the logical unit is associated.

Dynamic Change of Operands

You can dynamically change the following switched major node operands by editing the VTAMLST and then issuing the VARY ACT,UPDATE=ALL command. You can specify the new operand value on a higher-level definition statement for sifting, if applicable.

ADDR
ANS
ASDP
ASLENT
ASLTAB
BATCH
CALL
CAPACITY
CONNTYPE
COSTBYTE
COSTTIME
CPCP
DATMODE
DIALNO
DISCNT
DLCADDR
DLOGMOD
DLURNAME
DYNADJCP
DYNLU
ENCR
ENCRTYPE
FASTPASS
FEATUR2
GID
GRPNM
IDBLK
IDNUM
IPRESOLV
IRETRY
ISTATUS
LIMRES
LINENM
LOCADDR
LOGAPPL
LOGTAB
LUGROUP
LUSEED
MAXDATA
MAXOUT
MAXSESS
MDLENT
MDLTAB
MODETAB
NN
PACING
PASSLIM
PDELAY
PID
PUTYPE
REDIAL
REGISTER
SECURITY
SECNET
SHM
SHMTIM
SSCPFM
TERM
TGP
TOPO
UPARM1
UPARM2
UPARM3
USE
USSTAB
VERALSID
VERID
VERIFY
VPACING

In addition to the above VTAM operands, you can also dynamically change the following NCP operands, which are passed to the NCP during switched PU connection. For information on these operands, refer to the *NCP, SSP, and EP Resource Definition Reference*.

- COMRATE
- DLCI
- DYNWIND
- HPRQLIM
- INTFACE
- IPQLIM
- LADDR
- SNETMASK
- T1TIMER
- T2TIMER

To dynamically change an operand on the PU or LU statement, the resource to which it applies must be inactive. For instance, an operand on an LU definition statement that receives its value from a PU definition statement can be dynamically changed only when the LU is inactive; the PU can remain active.

You can dynamically change the operands associated with a PATH statement or tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) when the PU is active, connectable, or inactive. You can not change the PATH operands when the PU is in pending state.

Refer to *OS/390 SecureWay Communications Server: SNA Operation* and the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on the VARY ACT,UPDATE=ALL command.

Note: For a switched major node, you cannot add APPN capability or SDDLU support to a PU with the dynamic change function. If a PU has not been coded as APPN-capable, you cannot add the APPN operands to it. Likewise, if you have not coded LUGROUP (which allows SDDLU support) on a PU definition statement, you cannot add it with dynamic change.

NODETYPE Start Option

Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

The NODETYPE start option allows APPN function. The combination of NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

XCA and NCP Connections

Full Syntax

The full syntax for the switched major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=SWNET──────────────────────────────────►
    └─name─────┘

►──┬──────────────────────────────────────────────────────────────────►
   ├─CONFGDS=confgds_name──────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────┘

   ┌─VTAMTOPO=REPORT──────┐
►──┼──────────────────────┼──────────────────────────────────────────►◄
   │                      │
   └─VTAMTOPO=─┬─REPORT───┬─┘
               ├─NOREPORT─┤
               ├─INCLUDE──┤
               └─IGNORE───┘
```

Switched Major Node

GROUP

```
►►─┬──────┬─GROUP─┬──────────────────────────────┬──────────────────────────►◄
   └name──┘       └─VTAMTOPO=─┬─REPORT───┬─┘
                               └─NOREPORT─┘
```

PU

```
►►─name─PU─ADDR=link_station_address─┬─ANS=STOP──────────────┬──────────────►
                                     └─ANS=─┬─STOP────┬─┘
                                            └─CONTINUE─┘

                                                          (1)
►─┬─ASDP=NO──────┬─┬─AUTHLEN=YES──────┬─┬─CAPACITY=25M──────────┬──────────►
  └─ASDP=─┬─NO──┬─┘ └─AUTHLEN=─┬─NO──┬─┘ ├─CAPACITY=8K─────────┤
          └─YES─┘              └─YES─┘   └─CAPACITY=number_of_bits─┘

►─┬─CONNTYPE=value_of_conntype_start_option──────────────────────────────┬──►
  │ └─CONNTYPE=─┬─value_of_conntype_start_option─┬─┘                    │
  │             ├─APPN──────────────────────────┤                      │
  │             └─LEN───────────────────────────┘                      │
  │ ┌─CONNTYPE=APPN─┐  (2)                                             │
  └─┤               ├──────────────────────────────────────────────────┘
    └─CONNTYPE=APPN─┘

►─┬─COSTBYTE=0──────────────┬─┬─COSTTIME=0──────────────────┬──────────────►
  └─COSTBYTE=cost_per_byte─┘ └─COSTTIME=cost_per_unit_time─┘

►─┬─CPCP=value_of_cpcp_start_option──────────────────────────┬──────────────►
  └─CPCP=─┬─value_of_cpcp_start_option─┬─┘
          ├─NO───────────────────────┤
          └─YES──────────────────────┘

                                                 ┌─DATMODE=HALF──────┐
►─┬─CPNAME=control_point_name──────────┬─┬──────┴──────────────────┴──────►
  └─IDBLK=id_block,IDNUM=id_number─────┘ └─DATMODE=─┬─HALF──┬─┘
                                                    └─FULL──┘
```

```
►─┬─DISCNT=(NO)──────────────────────────────────────────┬─┬─DWACT=NO──────────┬──►
  │                                                      │ │                   │
  └─DISCNT=─┬─(NO)──,F──────────────────────────────────┬┘ └─DWACT=─┬─NO──┬────┘
            │─(YES─┬──────┬─)──────────────────────────┤            └─YES─┘
            │      └─,NF─┘                             │
            └─(DELAY─┬─,F──┬──┬──────────────────────┬─)┘
                     ├─,NF─┤  └─,delay_time──────────┘
                     └─,───┘
```

```
►─┬─DYNADJCP=value_of_dynadjcp_start_option──────────┬──────────────────────────►
  │                                                  │
  │ └─DYNADJCP=─┬─value_of_dynadjcp_start_option─┬──┤
  │             ├─NO──────────────────────────────┤  │
  │             └─YES─────────────────────────────┘  │
  │─┬─DYNADJCP=YES──┬  (3)                           │
  │ │               │                                │
  └─└─DYNADJCP=YES──┘                                └
```

```
►─┬─DYNLU=value_of_dynlu_start_option──────────┬─┬─ENCR=NONE──────────┬──────►
  │                                            │ │                    │
  │ └─DYNLU=─┬─value_of_dynlu_start_option─┬──┤ └─ENCR=─┬─NONE──┬────┘
  │          ├─NO───────────────────────────┤  │         ├─OPT───┤
  │          └─YES──────────────────────────┘  │         ├─REQD──┤
  │─┬─DYNLU=YES──┬  (4)                        │         └─SEL───┘
  │ │            │                             │
  └─└─DYNLU=YES──┘                             └
```

```
►─┬─ENCRTYPE=DES────┬─┬─HPR=YES──────────┬─┬──────────────────────────────────►
  │                  │ │                  │ │
  └─ENCRTYPE=TDES24──┘ └─HPR=─┬─NO──┬────┘ └─IDBLK=id_block──────────────────┘
                              └─YES─┘
```

```
►─┬──────────────────────────┬─┬─INCLUDOE=NO──────────┬─┬─IRETRY=NO──────────┬──►
  │                          │ │                      │ │                    │
  └─IDNUM=id_number──────────┘ └─INCLUDOE=─┬─NO──┬────┘ └─IRETRY=─┬─NO──┬────┘
                                           └─YES─┘                  └─YES─┘
```

```
►─┬─ISTATUS=ACTIVE──────────┬─┬─LIMRES=NO──────────┬─┬─LLERP=NOTPREF──────────┬──►
  │                         │ │                    │ │                        │
  │ └─ISTATUS=─┬─ACTIVE───┬──┤ └─LIMRES=─┬─NO──┬────┤ └─LLERP=─┬─REQUIRED──┬────┘
  │            └─INACTIVE─┘  │           └─YES─┘    │          └─NOTPREF───┘
  │                          │─┬─LIMRES=NO──┬  (5)  │
  │                          │ │            │       │
  └                          └─└─LIMRES=NO──┘       └
```

Switched Major Node

```
►──┬──────────────────────────────────────────────────────────────────────►
   ├─LUGROUP=lu_group_name──────────────────────────────────────────────┤
   └─LUGROUP=lu_group_name,LUSEED=lu_pattern_name───────────────────────┘

                 (7)
   ┌─MAXDATA=265──────────────────┐
                 (6)
   ├─MAXDATA=261──────────────────┤   ┌─MAXOUT=1──────────────────┐
►──┤                              ├───┤                           ├──────────►
   └─MAXDATA=number_of_bytes──────┘   └─MAXOUT=max_number_of_pius─┘

   ┌─MLTGPRI=0──────────┐
►──┤                    ├──┬──────────────────────────────────────────────►
   └─MLTGPRI=priority───┘  ├─NATIVE=──┬─NO──┬──┤  ├─NETID=network_id──┤
                           │          └─YES─┘  │  │
                           │                   │  │
                           │                   │  │
                           │                   
```

```
►─┬─TGCONF=(MULTI,NORMAL)──────────────────────────────────────────────►
  │
  └─TGCONF=(──┬─────────┬──,──┬──────────────────────┬──)─┘
              ├─MULTI───┤     ├─NORMAL───────────────┤
              └─SINGLE──┘     └─STANDBY──,──┬─────┬──┘
                                            ├─nn──┤
                                            └─NONE┘

                 (10)
  ┌─TGN=ANY──────────┐
  │       (9)        │
  ├─TGN=1────────────┤                    ┌─TOPO=CONNECT──────────┐
►─┤                  ├──┬──────────────┬──┤                       ├──►
  │        ┌─ANY────┐│  └─TGP=tgp_name─┘  │        ┌─CONNECT────┐│
  │        ├─1──────┤│                    └─TOPO=──┤            ├┘
  └─TGN=───┤         ├┘                           ├─connection_cost┤
           └─tg_number┘                           └─NONE───────────┘

►─┬─UPARM1=128──────────────────────┬──┬─UPARM2=128──────────────────────┬──►
  │                                 │  │                                 │
  └─UPARM1=user-defined_operand_value┘  └─UPARM2=user-defined_operand_value┘

►─┬─UPARM3=128──────────────────────┬──┬─VERALSID=NO─────────┬──────────►
  │                                 │  │                     │
  └─UPARM3=user-defined_operand_value┘  │          ┌─NO──┐    │
                                        └─VERALSID=┤     ├────┘
                                                   └─YES─┘
                                        ┌─VERALSID=NO──┐ (11)
                                        └─VERALSID=NO──┘

                                   ┌─XNETALS=value_of_XNETALS_start_option─┐
►──┬──────────────────────────────┬──┬──────────────────────────────────────┬──►◄
   │              ┌─REPORT───┐    │  │                  ┌─NO──┐             │
   └─VTAMTOPO=────┤          ├────┘  └─XNETALS=──────────┤     ├─────────────┘
                  └─NOREPORT─┘                           └─YES─┘
```

Notes:

1 Capacity defaults to 25M for ATM native connections.

2 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

3 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

4 DYNLU=YES is the only valid value if the TRLE operand is coded.

5 LIMRES=NO is the only valid value if the TRLE operand is coded.

6 MAXDATA defaults to 261 when PUTYPE=1 is coded.

7 MAXDATA defaults to 265 when PUTYPE=2 is coded.

8 NEGLIGIB is the default if the TRLE operand is coded.

9 TGN defaults to 1 for subarea.

10 TGN defaults to ANY for APPN.

11 VERALSID=NO is the only valid value if the TRLE operand is coded.

PATH

```
►►──┬──────────┬──PATH──┬─CALL=INOUT──────────┬──────────────────────────────────────►
    └─name─────┘        │                     │
                        └─CALL=──┬─INOUT──┬──┘
                                 ├─IN─────┤
                                 └─OUT────┘

►──┬─DIALNO=dial_digits──────────────────────────────────────────────────────────────►
   ├─DLCADDR=(─subfield_id,type,─string──)──────────────────────────────────────────┤
   ├─DIALNO=dial_digits,─DLCADDR=(─subfield_id,type,─string──)──────────────────────┤
   └─LINENM=line_name──────────────────────────────────────────────────────────────┘

►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──────►
   └─DLURNAME=dlur_name───┘  └─GID=group_identifier─┘  └─GRPNM=group_name────┘

►─LINENM=line_name──┬──────────────────────┬──┬─REDIAL=3──────────────────┬──────────►
                    └─PID=path_identifier──┘  └─REDIAL=number_of_redials──┘

►──┬─SHM=NO──────────┬──┬─SHMTIM=0.0──┬──┬─USE=YES─────────┬──────────────────────────►
   │                 │  └─SHMTIM=time─┘  │                 │
   └─SHM=──┬─NO──┬──┘                   └─USE=──┬─YES──┬──┘
           └─YES─┘                              └─NO───┘

►──┬──────────────────────────────────────┬──┬─VERIFY=NONE─────────────┬──────────►◄
   └─VERID=──┬─call_identifier──────────┬──┘  │                         │
             └─(─call_identifier──────)─┘     └─VERIFY=──┬─NONE──┬──┘
                                                         ├─IN────┤
                                                         ├─INOUT─┤
                                                         └─OUT───┘
```

LU

```
►►─name──LU──LOCADDR=address──┬──────────────────────────────────────┬──────────────►
                              └─ASLENT=associated_lu_table_entry_name─┘

►──┬──────────────────────────────────┬──┬─BATCH=NO────────────┬──────────────────►
   └─ASLTAB=associated_lu_table_name──┘  │                     │
                                         └─BATCH=──┬─NO──┬──┘
                                                   └─YES─┘
```

```
►─┬─CERTIFY=NO──────────────┬─┬─CKEYNAME=name──────────────┬─┬─CLRSESSQ=NO──────────────┬──►
  │                         │ │                            │ │                          │
  └─CERTIFY=─┬─NO──┬─────────┘ └─CKEYNAME=crypto_key_name────┘ └─CLRSESSQ=─┬─NO──┬────────┘
             └─YES─┘                                                      └─YES─┘

►─┬──────────────────────────────────────────────────────────────────────────────────────
  │
  └─DLOGMOD=default_logon_mode_table_entry_name──┬─┬─ENCR=NONE──────────────┬──────────►
                                                 │ │                        │
                                                 └─ENCR=─┬─NONE──┬──────────┘
                                                         ├─OPT───┤
                                                         ├─REQD──┤
                                                         └─SEL───┘

►─┬─ENCRTYPE=DES───────┬─┬─FASTPASS=YES──────────┬──────────────────────────────────────►
  │                    │ │                       │
  └─ENCRTYPE=TDES24────┘ └─FASTPASS=─┬─YES──┬─────┘
                                     └─NO───┘

►─┬─FEATUR2=(DUALCSE,MODEL1,NOEDATS,NOPRINTR,NOSELPEN)──────────────────────────────────►
  │
  └─FEATUR2=(──┬─┬─DUALCSE──┬─┬──────────────────────────────────────────────────────────
               │ │          │ │
               │ └─LOWERCSE─┘ │
               │ ┌─MODEL1───┐ │
               │ │          │ │
               │ └─MODEL2───┘ │
               │ ┌─NOEDATS──┐ │
               │ │          │ │
               │ └─EDATS────┘ │
               │ ┌─NOPRINTR─┐ │
               │ │          │ │
               │ └─PRINTR───┘ │
               │ ┌─NOSELPEN─┐ │
               │ │          │ │
               └─└─SELPEN───┘─┘──)──────────────────────────────────────────────────────

►─┬─ISTATUS=ACTIVE──────────┬──┬──────────────────────────────────────────────────────►
  │                         │  │
  └─ISTATUS=─┬─ACTIVE──┬────┘  └─LOGAPPL=controlling_primary_lu──────────────────────────
             └─INACTIVE─┘

►─┬─────────────────────────────────┬─┬─────────────────────────────────────────────────►
  │                                 │ │
  └─LOGTAB=interpret_table_name─────┘ └─MAXSESS=max_number_LU-LU_sessions─────────────────

►─┬─────────────────────────────────────────┬─┬─────────────────────────────────────────►
  │                                         │ │
  └─MDLENT=model_name_table_entry_name──────┘ └─MDLTAB=model_name_table_name──────────────
```

Switched Major Node

```
►─┬─MODETAB=ISTINCLM──────────────────┬─┬─PACING=1──────────────────┬──────────────►
  └─MODETAB=logon_mode_table_name─────┘ │                           │
                                        └─PACING=─┬─1──────────────┬─┘
                                                  └─pacing_number──┘

                   (2)
  ┬─REGISTER=NO──────────────────┬
                        (1)      │
  ├─REGISTER=NETSRVR─────────────┤   ┬─RESETLU=YES──────────┬──────────────────────►
  │                              │   │                      │
  │                 ┬─NO──────┬  │   └─RESETLU=─┬─YES──┬──┘
  │                 ├─NETSRVR─┤  │              │      │
  └─REGISTER=───────┤         ├──┘              └─NO───┘
                    └─CDSERVR─┘

  ┬─RESSCB=0──────────────────────────────┬──┬──────────────────────────┬──────────►
  └─RESSCB=number_of_session_control_blocks┘  └─SPAN=span_name──────────┘

  ┬─SSCPFM=FSS──────────┬──┬──────────────────────────────────────────────────────►
  │                     │  │
  │                 ┬─FSS─────┬  │  ┬─TERM=─┬─3275──────────┬──┬
  └─SSCPFM=─────────┤         ├──┘  │       ├─3277──────────┤  │
                    ├─USSNTO──┤     │       ├─3284──────────┤  │
                    ├─USSSCS──┤     │       ├─3286──────────┤  │
                    ├─USS3270─┤     │       └─terminal_Type─┘  │
                    ├─USS3275─┤     │                          │
                    └─USS3780─┘     └──────────────────────────┘

  ┬─USSTAB=ISTINCDT──────────────┬─┬─VPACING=2──────────────────┬──────────────►◄
  └─USSTAB=uss_table_name────────┘ │                            │
                                   └─VPACING=─┬─2──────────────┬─┘
                                              └─number_of_requests─┘
```

Notes:

1 REGISTER defaults to NETSRVR for dependent LUs.

2 REGISTER defaults to NO for independent LUs.

Enterprise Extender Connections

The full syntax for the switched major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►─┬──────────┬─VBUILD──TYPE=SWNET──────────────────────────────────────────────►
   └─name─────┘
```

```
►──┬──────────────────────────────────────────────────────────────────►
   ├─CONFGDS=confgds_name──────────────────────────────────────────────┤
   └─CONFGPW=confgds_password,CONFGDS=confgds_name─────────────────────┘

   ┌─VTAMTOPO=REPORT──────────┐
►──┼──────────────────────────┼──────────────────────────────────────►◄
   │                ┌─REPORT──┐│
   └─VTAMTOPO=──────┼─────────┼┘
                    ├─NOREPORT─┤
                    ├─INCLUDE──┤
                    └─IGNORE───┘
```

PU

```
                   ┌─ISTATUS=ACTIVE──────────┐
►►──name──PU───────┼─────────────────────────┼──┬──────────┬──┬──────────────────────────┬──►
                   │                ┌─ACTIVE─┐│  └─PUTYPE=2─┘  └─SPAN=span_name──────────┘
                   └─ISTATUS=───────┼────────┼┘
                                    └─INACTIVE┘

            (1)
   ┌─TGN=ANY──────────────┐
►──┼──────────────────────┼──────────────────────────────────────────►◄
   │              ┌─ANY──┐│
   └─TGN=─────────┼──────┼┘
                  └─tg_number┘
```

Notes:

1 TGN defaults to ANY for APPN.

```
                                   ┌─XNETALS=value_of_XNETALS_start_option─┐
►►──┬──────────────────────────────┼───────────────────────────────────────┼──────────────────►◄
   │              ┌─REPORT──┐      │ XNETALS=──┬─NO──┬──────────────────────┘
   └─VTAMTOPO=────┼─────────┼──────┘           └─YES─┘
                  └─NOREPORT┘
```

PATH

```
                   ┌─CALL=INOUT──────────┐
►►──┬──────────┬───┼─────────────────────┼──┬──────────────────────────┬──────────────────►
   └─name─────┘   │              ┌─INOUT─┐│  └─DLURNAME=dlur_name──────┘
   PATH           └─CALL=────────┼───────┼┘
                                 ├─IN────┤
                                 └─OUT───┘

►──┬──────────────────────────┬──┬──────────────────────────┬──────────────────────────────►
   └─GID=group_identifier─────┘  └─GRPNM=group_name──────────┘
```

Switched Major Node

```
►──┬──────────────────────────────────┬──┬──────────────────────────────────┬──►
   └─HOSTNAME=remote_host_name (1)────┘  └─IPADDR=address (1)──────────────┘

   ┌─IPRESOLV=(45)──────┐
►──┼────────────────────┼──LINENM=line_name──┬──────────────────────────┬──►
   └─IPRESOLV=(1-255)───┘                     └─PID=path_identifier──────┘

   ┌─REDIAL=3──────────────────┐  ┌─SAPADDR=4──────┐  ┌─USE=YES──────────┐
►──┼───────────────────────────┼──┼────────────────┼──┼──────────────────┼──►
   └─REDIAL=number_of_redials──┘  └─SAPADDR=address┘  │      ┌─YES──┐    │
                                                      └─USE=─┼──────┼────┘
                                                             └─NO───┘

                                                   ┌─VERIFY=NONE──────────┐
►──┬──────────────────────────────────┬──────────┼──────────────────────┼──►◄
   │                                  │          │          ┌─NONE──┐   │
   └─VERID=──┬─call_identifier──────┬─┘          └─VERIFY=──┼───────┼───┘
             └─(─call_identifier──)─┘                       ├─IN────┤
                                                            ├─INOUT─┤
                                                            └─OUT───┘
```

Notes:

1. HOSTNAME and IPADDR are mutually exclusive, however, IPADDR is used if both are specified.

Local (Source) and Remote (Destination) Service Access Point Detemination

The local (source) service access points (SAPs) and remote (destination) SAPs are determined as follows:

SAP Type	Predefined TGs	Connection Network
Local SAP	XCA major node SAPADDR keyword on PORT statement. Default is 4.	XCA major node, SAPADDR on PORT statement.
Remote SAP	Switched major node, SAPADDR keyword on PATH statement. Default is 4.	Learned during connection.

Notes:

1. These SAP pairs are used in LDLC signals.
2. For more information on SAP pairs refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Operand Descriptions

The following section describes the operands you can code for the switched major node.

ADDR

```
►►──ADDR=link_station_address──────────────────────────────────────────►◄
```

statements: *PU*
dependencies: *nonswitched connections*

Specifies the hexadecimal SDLC station address for the physical unit. You can specify one or two hexadecimal digits for ADDR. For example, if the address is X'03', you can code ADDR=03 or ADDR=3.

This address is required for all physical units attached through SDLC lines and X.25 networks.

This operand is ignored for switched subarea connections.

ANS

```
                ┌─ANS=STOP──────────┐
►►──────────────┼──────────────────┼──────────────────────────────────►◄
                │          ┌─STOP──┐│
                └─ANS=─────┼───────┼┘
                           └─CONTINUE┘
```

statements: *GROUP, PU*
dependencies: *none*

Note: This operand is valid for subarea and PU Type 2 connections.

Determines whether the physical unit continues to operate when the NCP enters automatic network shutdown. The value coded on this statement overrides whatever value you code on the ANS definition statement for the NCP.

For switched subarea connections, this operand specifies whether a switched subarea connection over lines attached to this physical unit should remain active or be dropped when the owning SSCP is lost. This is used only on a PU definition statement that defines a subarea physical unit that is connected to a link station within an NCP.

ANS=CONTINUE (or CONT)
Indicates that the LU-LU sessions are to continue.

Code ANS=CONTINUE if the terminal participates in XRF sessions.

When a session is continued and takeover occurs, if the LU taken over is a dependent logical unit, it must support ACTLU(ERP) or the session is terminated during takeover processing. An independent logical unit does not receive ACTLU, so this restriction does not apply.

Note that there are several restrictions for session continuity in certain configurations. For information on these restrictions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

ANS=STOP
Indicates that the connection should be dropped.

ASDP

```
►►─┬─ASDP=NO──────┬──────────────────────────────────────────►◄
   │              │
   └─ASDP=─┬─NO──┬─┘
           └─YES─┘
```

statements:	*GROUP,PU*
dependencies:	*none*

Specifies whether an application can supply dial information and other parameters for a dial-out switched connection for this PU.

ASDP=NO
Indicates that an application cannot supply dial information.

ASDP=YES
Indicates that an application can supply dial information.

ASLENT

```
►►─┬──────────────────────────────────────────┬──────────────►◄
   └─ASLENT=associated_lu_table_entry_name────┘
```

statements:	*GROUP, PU, LU*
dependencies:	*ASLTAB; PUTYPE=1,2*

Specifies the name of the associated LU table entry to be used for this logical unit.

If you code the ASLENT operand, you must also code the ASLTAB operand. If you omit ASLENT and include ASLTAB, VTAM uses the first entry in the associated LU table by default.

ASLTAB

```
►►─┬──────────────────────────────────────────┬──────────────►◄
   └─ASLTAB=associated_lu_table_name──────────┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*

Specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 661 for a description of how to define an associated LU table.

If you omit the ASLTAB operand, VTAM will not provide the names of associated logical units to the application during LU-LU session initiation. However, you can provide the associated LU names in other ways. You can enter them from the terminal on the PRINTER1 and PRINTER2 operands of the LOGON command, or

VTAM can send them to the application in a formatted session-initiation request. Refer to “LOGON Command” on page 758 for more information on the PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

AUTHLEN

```
►►─┬─AUTHLEN=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─AUTHLEN=─┬─NO──┬──────┘
              └─YES─┘
```

statements:	*GROUP,PU*
dependencies:	*Type 2.1; APPN only*

Specifies whether VTAM should pass the transmission priority field specified by this PU to another PU.

AUTHLEN=NO
VTAM will not pass the specified transmission priority, instead the default priority of medium will be passed.

AUTHLEN=YES
VTAM will pass the specified transmission priority.

BATCH

```
►►─┬─BATCH=NO─────────────┬──────────────────────────────────────────►◄
   │                      │
   └─BATCH=─┬─NO──┬───────┘
            └─YES─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2; pre-V4R3 NCPs only*

Specifies the processing priority that the NCP uses for the logical unit.

BATCH=NO
BATCH=NO means a high priority (suitable for interactive application programs).

BATCH=YES
BATCH=YES means a low priority.

Note: This operand applies to pre-V4R3 NCPs only. If you code the BATCH operand and the NCP is V4R3 or later, the operand is ignored.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information about this operand.

CALL

```
   ┌─CALL=INOUT──────┐
►►─┤                 ├──────────────────────────────────────────────►◄
   │          ┌─INOUT─┐
   └─CALL=────┤       ├
              ├─IN────┤
              └─OUT───┘
```

statements: PATH
dependencies: GRPNM

Specifies whether inbound and outbound calls are to be supported by this PATH definition statement. If all PATH definition statements for this physical unit specify CALL=INOUT, an inbound call is accepted from any switched line group. If you specify call selection (CALL=IN or CALL=OUT) on any PATH definition statement, an inbound call is accepted only if it is using a line group for which there is a PATH definition statement that specifies CALL=IN or CALL=INOUT. Therefore, if one PATH definition statement specifies call selection, all valid PATHs (inbound and outbound) must be defined.

If you specify CALL, GRPNM is required.

If CALL=OUT or CALL=INOUT, either DIALNO or LINENM is required.

CALL=IN
Specifies that inbound calls can be accepted but outbound calls cannot be made using this PATH definition statement.

CALL=INOUT
Specifies that inbound calls can be accepted and outbound calls can be made using this PATH definition statement.

CALL=OUT
Specifies that outbound calls can be made but no inbound calls can be accepted using this PATH definition statement.

Note: See the description of the VERIFY operand if you specify both CALL and VERIFY.

CAPACITY

```
                 (1)
   ┌─CAPACITY=25M────────────┐
   ├─CAPACITY=8K─────────────┤
►►─┤                         ├──────────────────────────────────────────►◄
   └─CAPACITY=number_of_bits─┘
```

Notes:

1 Capacity defaults to 25M for ATM native connections.

statements: *GROUP, PU*
dependencies: *APPN only*

Specifies the effective capacity of the link that comprises the transmission group (TG). Specify the value in either Kb per second (for example, 100K) or Mb per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Notes:

1. Because the value for capacity is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed, as the same value. See Table 44 on page 515 for a list of CAPACITY values you can specify and their corresponding values when displayed.
2. The capacity specified for ATM native connections defined as reserved bandwidth should be related to the forward sustainable cell rate (DLCADDR subfield 7). To calculate the data throughput required for a reserved bandwidth connection, multiply the sustainable cell rate by the payload per cell (48 bytes). If the defined CAPACITY is greater than ten times the ATM reserved bandwidth connection capacity, excess storage might be utilized and the weighting of this TG with others might cause undesirable results.

CERTIFY

```
►►─┬─CERTIFY=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─CERTIFY=─┬─NO──┬────┘
              └─YES─┘
```

statements: *PU, LU*
dependencies: *CERTIFY is ignored unless ENCR is also active. Before enabling the CERTIFY function, ensure that the device or software providing the partner's LU support also supports the CERTIFY function.*

Specifies whether VTAM on behalf of the application program will verify authentication.

CERTIFY=YES
Specifies SLU authentication (verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CERTIFY=NO
Specifies no SLU authentication (not verifying the SLU is using the same cryptographic key as the PLU) is to be performed by the PLU.

CKEYNAME

```
►►─┬─CKEYNAME=name──────────────┬──────────────────────────────────►◄
   │                            │
   └─CKEYNAME=crypto_key_name───┘
```

statements: *LU*
dependencies: *dependent LUs only*

Specifies the name of the key-encrypting key (KEK) in the cryptographic key data set (CKDS) for the LU. Use CKEYNAME to encipher the session key that is sent in the BIND to the LU when a cryptographic session is established. CKEYNAME has the same format and restrictions as other VTAM *names*. For information on naming conventions, see “Format of Definition Statements” on page 9.

If you do not specify CKEYNAME, the cryptographic key name defaults to the name on the LU definition statement.

Note: Independent LUs will ignore the CKEYNAME keyword and use their default name.

CLRSESSQ

```
►►─┬─CLRSESSQ=NO──────────┬──────────────────────────────────────►◄
   │                      │
   └─CLRSESSQ=─┬─NO──┬────┘
               └─YES─┘
```

statements: *LU*

Specifies whether a dependent LU is forced into an inhibited session state when the LU is disabled. When the LU is enabled, it is returned to normal status. Sessions, however, are not connected.

CLRSESSQ=YES
Specifies that the LU is forced into an inhibited session state. Both active and pending sessions are removed from the session queue.

CLRSESSQ=NO
Specifies that the LU is not forced into inhibited session state. This is the default.

CONFGDS

```
►►──┬──────────────────┬──────────────────────────────────────►◄
    └─CONFGDS=name─────┘
```

statements: *VBUILD*
dependencies: *none*

Specifies a 1–8 character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement that has this data definition name in the VTAM start procedure.

CONFGPW

```
►►──┬──────────────────┬──────────────────────────────────────►◄
    └─CONFGPW=password─┘
```

statements: *VBUILD*

dependencies: *CONFGDS*

Specifies the 1–7 character alphanumeric password that VTAM uses to access the configuration restart data set. If you do not code CONFGPW, and it is required by VSAM, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

Code CONFGPW only if you also code CONFGDS.

CONNTYPE

```
      ┌─CONNTYPE=value_of_conntype_start_option──────┐
►►────┼──────────────────────────────────────────────┼──────────────────────────────►◄
      │              ┌─value_of_conntype_start_option─┐│
      └─CONNTYPE=────┼────────────────────────────────┼┘
                     ├─APPN───────────────────────────┤
                     └─LEN────────────────────────────┘
      ┌─CONNTYPE=APPN─┐  (1)
      │               │
      └─CONNTYPE=APPN─┘
```

Notes:

1 CONNTYPE=APPN is the only valid value if the TRLE operand is coded.

statements: *GROUP, PU*
dependencies: *APPN only; NODETYPE start option*

Specifies for a type 2.1 PU whether the connection is to be established as a LEN connection or attempted as an APPN connection.

If you do not code the NODETYPE start option, the CONNTYPE operand is ignored and the connection will be a LEN connection.

CONNTYPE=APPN
Specifies that this connection can support parallel TGs, CP-CP sessions, and CP name change support. CONNTYPE=APPN must be coded for CPCP=YES to be valid.

CONNTYPE=LEN
Specifies that this connection does not support parallel TGs, CP-CP sessions, or CP name change support.

Note: For LEN sessions between PUs when CONNTYPE=APPN is coded on each PU, the CPNAME of each PU must be unique. When CONNTYPE=LEN, the CPNAMEs can be the same.

Since CONNTYPE=LEN indicates that the PU is not APPN-capable, do not code any APPN-only operands on the same PU statement. If an APPN-only operand is coded with CONNTYPE=LEN, VTAM will issue a message warning of a parameter conflict.

Although you need to code CONNTYPE=APPN for a PU to be an APPN resource, this does not necessarily mean the PU will be APPN; a PU's characteristics are determined at activation. However, coding CONNTYPE=LEN means that the PU can only be a LEN resource, even if, at activation, the XID indicates the PU can be an APPN resource.

COSTBYTE

```
►►─┬─COSTBYTE=0──────────────┬──────────────────────────────────────────►◄
   └─COSTBYTE=cost_per_byte──┘
```

statements:	*GROUP, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a cost-per-byte-transmitted to be associated with the transmission group (TG). A value of 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
►►─┬─COSTTIME=0──────────────────┬──────────────────────────────────────►◄
   └─COSTTIME=cost_per_unit_time─┘
```

statements:	*GROUP, PU*
dependencies:	*APPN only; TOPO*
range:	*0–255*

Specifies a cost-per-unit-of-time to be associated with the transmission group (TG). A value of 0 is the least expensive cost per unit of time and 255 is the most expensive.

If TOPO is coded with a connection cost specified, then the value specified through TOPO will be used as the cost per unit time until the connection is established. The COSTTIME value specified is used while the connection is active.

CPCP

```
►►─┬─CPCP=value_of_cpcp_start_option──────────┬──────────────────────────►◄
   │                                          │
   └─CPCP=─┬─value_of_cpcp_start_option─┬──────┘
           ├─NO───────────────────────┤
           └─YES──────────────────────┘
```

statements:	*GROUP, PU*
dependencies:	*APPN only*

Specifies whether CP-CP sessions are supported on this connection. If you do not code CPCP, the value defaults to YES only if you specify YES on the CPCP start option. Otherwise, the value defaults to NO.

CPCP=NO
CP-CP sessions are not supported for this connection.

CPCP=YES
CP-CP sessions are supported for this connection.

Note: CPCP=YES is valid only when CONNTYPE=APPN.

CPNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └CPNAME=control_point_name─┘
```

statements:	*PU*
dependencies:	*Type 2.1*

Specifies the control point (CP) name of a type 2.1 peripheral node. If a type 2.1 peripheral node supports it, coding CPNAME is required if the peripheral node supports role negotiation (such as 3172, 3174, and Communications Manager).

To dial in, a type 2.1 peripheral node on a switched line requires either CPNAME or both IDBLK and IDNUM on the PU definition statement. However, you can code all three operands.

To dial out without knowing the partner's CPNAME or IDBLK and IDNUM, code CPNAME without a value (CPNAME=).

For dial-in or dial-out, if CPNAME is not in the XID or if you do not code CPNAME, VTAM uses IDBLK and IDNUM instead.

VTAM uses the CP name of a type 2.1 PU to identify the PU in any messages. For an APPN PU, its *name* (or label) must differ from *control_point_name* (specified here on CPNAME) to ensure that a CP-CP session establishment succeeds. If *name* and *control_point_name* are duplicates, the session establishment fails and sense code 08880001 is given.

When VTAM is defined as a type 2.1 node, code CPNAME the same as SSCPNAME on the other side of the connection. Ensure that SSCPNAME is unique within the host domain.

Refer to the appropriate product manual for CPNAME, IDNUM, and IDBLK values.

DLUR physical units: For DLUR-defined PUs, CPNAME may be specified for a single PU which is non-adjacent to the DLUS (non-adjacent means there is no APPN connection between the DLUS and the control point named through CPNAME). When the DLUR-defined PU is adjacent to the DLUS, or when more than one PU with the same CPNAME must be defined, a different means of identification dependent upon DLC support must be used (for example, IDBLK/IDNUM for internal PUs). Additional information can be found in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and in the publications for your DLUR implementation.

DATMODE

```
    ┌─DATMODE=HALF──────┐
►►──┼───────────────────┼──────────────────────────────────────────────►◄
    │            ┌─HALF─┐│
    └─DATMODE=───┼──────┼┘
                 └─FULL─┘
```

statements: *GROUP, PU*
dependencies: *none*

Specifies whether the physical unit communicates in half-duplex or full-duplex data mode. The associated line must support simultaneous transmission in both directions.

DATMODE=FULL
Indicates that the physical unit communicates in full-duplex mode.

DATMODE=HALF
Indicates that the physical unit communicates in half-duplex mode.

DIALNO

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─DIALNO=dial_digits─┘
```

statements: *PATH*
dependencies: *LINENM*

Specifies the dial information used to initiate a connection with a physical unit over a switched link.

If you do not code LINENM, DIALNO is required if this PATH definition statement is for a type 4 or 5 physical unit.

DIALNO is mutually exclusive with the LINENM operand for any single switched path definition statement.

Public Telephone Networks: DIALNO consists of a combination of digits and special characters. If an auto-call unit is installed, the special characters consist of separator characters, dialing pause characters, and end-of-number characters, as follows:

- The separator character causes the auto-call unit to delay sending the next dial digit until it receives a secondary dial tone. For separator characters, use underscores (_) or the hexadecimal equivalent (X'6D').
- If the auto-dial unit is not equipped to use separator characters, a dialing-pause character can be used to allow time to receive a secondary dial tone. A dialing-pause character pauses for a predetermined interval before sending the next digit. You can insert as many vertical bars (|) or the hexadecimal equivalent (X'FA') as you need to denote dialing pauses.
- Some modems require end-of-number characters. For end-of-number characters, use
  - asterisks (*) or the hexadecimal equivalent (X'5C')
  - percent signs (%) or the hexadecimal equivalent (X'6C')
  - at-signs (@) or the hexadecimal equivalent (X'7C')

  For example, you could code DIALNO=8_5799*, where _ is the separator character and * is the end-of-number character.
- The maximum length of the operand is 32 characters, including vertical bars, the separator character, and the end-of-number character.

For more information on this operand, refer to the *NCP, SSP, and EP Resource Definition Reference*.

NCP/Token-Ring interconnection (NTRI):

Use the following format when coding DIALNO for a terminal that is called by a host through NCP/Token-Ring interconnection

```
DIALNO=aabbcccccccccccc
```

where:

aa The token-ring interface coupler (TIC) number of the communication controller (a decimal number in the range 00–99). The TIC number equates with the PORTADD keyword on the NCP LINE definition statement.

bb or Xbb
The service access point (SAP) address of the terminal. The SAP address must be a multiple of 4, and can be a hexadecimal or decimal number.

cccccccccccc
The last 6 bytes of the terminal's ring-station address (the first digit must be in the range 4–7). The address must consist of 12 hexadecimal digits that range from X'4000 0000 0000' to X'7FFF FFFF FFFF' using the following rules:

- If bytes 0 and 1 are equal to either X'4000' or X'7FFF' then bytes 2 through 5 must be unique on the network.
- If bytes 0 and 1 are in the address range X'4001' through X'7FFE' then bytes 0 through 5 must be unique on the network.

Note: You cannot code 000000000000 for the last 12 digits of the DIALNO operand on the PATH definition statement.

This operand applies only if the physical unit is attached through NCP/Token-Ring interconnection. DIALNO is required if the physical unit is used as a dial-out physical unit.

Frame Relay Devices: Use the following format when coding DIALNO for dialing out to a frame relay device:

```
DIALNO=aabb00000000cccc
```

or

```
DIALNO=aaXbb00000000cccc
```

where:

aa The port number specified with the PORTADD keyword of the frame relay link, in decimal.

bb or Xbb
The remote station's service access point (SAP), in decimal (bb) or hexadecimal (Xbb).

cccc The data link connection identifier (DLCI) for the connection, in hexadecimal.

For frame relay BAN-attached devices, you must also specify the remote MAC address for the BAN station, in hexadecimal, as shown by **dddddddddddd** in the following format:

```
DIALNO=aabb00000000ccccdddddddddddd
```

or

```
DIALNO=aaXbb00000000ccccdddddddddddd
```

The MAC address must be in the range 1–X'7FFF FFFF FFFF'.

IBM 3172 Nways Interconnect Controller Switched Data Networks: Use the following DIALNO format for host-initiated connections to peripheral devices connected through the IBM 3172 Nways Interconnect Controller (*aa*, *bb*, and *cccccccccccc* are hexadecimal):

```
DIALNO=aabbcccccccccccc
```

where:

aa A 2 digit place-holder. It has no meaning in the IBM 3172 Nways Interconnect Controller environment.

bb The service access point (SAP) address of the terminal. It must be a multiple of 4.

cccccccccccc
The medium access control (MAC) address for the peripheral device on the local area network.

Notes:

1. You cannot code 000000000000 for the last 12 digits (MAC address) on the DIALNO operand on the PATH definition statement.
2. If the PU is attached to a LAN through the IBM 3172 Nways Interconnect Controller using an 8209 LAN bridge, the MAC address must be coded differently. Because the 8209 will flip the bits in every byte of the MACADDR, the MACADDR should be coded in inverted order in the DIALNO.
3. Certain levels of NCP do not support hexadecimal digits X'A'–X'F' as part of the MAC address.

DISCNT

```
►►─┬─DISCNT=(NO)──────────────────────────────────────────┬──────────────────────────────►◄
   │                                                      │
   └─DISCNT=─┬─(NO)──,F──────────────────────────────────┬─┘
             │                                           │
             ├─(YES──┬──────┬──)──────────────────────────┤
             │       └─,NF──┘                            │
             │                                           │
             └─(DELAY──┬─,F──┬──┬──────────────────┬──)──┘
                       │     │  │                  │
                       ├─,NF─┤  └─,delay_time──────┘
                       │     │
                       └─,───┘
```

statements:	*GROUP, PU*
dependencies:	*PUTYPE=2*

Specifies when VTAM should terminate its SSCP-LU and SSCP-PU sessions and, when DISCNT=YES or DISCNT=DELAY, whether to indicate “final-use” status in the DACTPU request unit when it deactivates a PU

DISCNT=(YES)

DISCNT=(NO)
Specifies whether VTAM physically disconnects the PU when the last LU ends its session with its application program (that is, when there are no more LU-LU sessions).

For a PU on a switched link, disconnection means that the dial connection is broken (in effect, the telephone is hung up, to save telephone charges) and the SSCP-PU session is terminated. Disconnection on a switched link does not involve deactivating the PU or its LUs (that is, sessions can be requested with those LUs causing the physical connection to be re-established).

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about disconnection.

DISCNT=(YES,F)

DISCNT=(YES,NF)
Specifies whether VTAM indicates “final-use” status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES. This value does not apply when DISCNT=NO, nor does it have any effect on the VARY INACT command.

If you code F or use the default, “final-use” status is indicated and the connection can be ended. If you code NF, “not-final-use” status is indicated and the connection should not be ended.

Each device has its own requirements regarding “final-use” status. To determine whether to code F or NF for a given device, consult the appropriate installation publication for the device.

DISCNT=(DELAY,F,*delay_time***)**

DISCNT=(DELAY,NF,*delay_time***)**

DISCNT=(DELAY,,*delay_time***)**
Specifies that VTAM disconnects the physical unit if no LU-LU sessions exist after the value specified for *delay_time* expires. If *delay_time* is not specified, the current value for the DISCNTIM start option is used. *delay_time* specifies the number of seconds that VTAM delays disconnecting this PU if no LU-LU sessions exist. A decimal value in the range 5–65535 is valid.

Note: A delay time value can also be assigned for this PU through the MODIFY DEFAULTS operator command.

For HPR Sessions: If this VTAM is a node in a high performance routing (HPR) route and is connected to an IBM 37xx Communication Controller, an IBM S/390 Open Systems Adapter or through an IP network using Enterprise Extender, NCP or VTAM monitors the session traffic on the link to determine whether to deactivate a PU. When DISCNT=(DELAY,,*delay_time*), NCP or VTAM disconnects the PU if no session traffic has occurred during the time interval specified by *delay_time*. When DISCNT=(DELAY) or DISCNT=(YES), NCP or VTAM disconnects the PU if no session traffic has occurred during the time interval specified on the DISCNTIM start option.

DLCADDR

```
►►──┬──────────────
```

dependencies: *LINENM*
range: *See description*

Specifies the signaling information used to initiate a connection with a physical unit over a switched link.

Multiple DLCADDR operands can be coded on a single PATH definition statement. For example, the following can be coded on a single PATH statement.

```
PATH  DLCADDR=(1,TYPE1,STRING1),
      DLCADDR=(2,TYPE2,STRING2)
```

subfield_id
Indicates the subfield. A value between 1–96 can be specified.

The first DLCADDR operand must have a *subfield_id* of 1. Also, each operand must have a unique *subfield_id*. If two DLCADDR operands—following a DLCADDR operand with a subfield of 1—have the same *subfield_id*, VTAM treats the two operands as duplicates. VTAM ignores the second operand and issues an error message. If a DLCADDR operand with a subfield of 1 immediately follows another DLCADDR operand with a subfield of 1, VTAM does not ignore the second operand. It treats the second operand as the beginning of another set of signaling information.

Note: If you receive an error message referring to the first DLCADDR operand coded on a PATH statement, limited syntax checking is done on the subsequent DLCADDR operands coded.

For DLUR PUs, refer to the documentation related to the DLUR implementer or to the device that supports the routing of the data link connection for information about what the specific values indicate. For ATM native connections, refer to the information on page 426 of this book for information about what the specific values indicate.

type
Indicates the data format of the signaling information and is one of the following:

Type	Data Format
A	Compressed alphanumeric
BCD	Binary coded decimal
C	EBCDIC
D	Decimal
I	ASCII
X	Hexadecimal

Type C (EBCDIC) is the default.

string
Defines the signaling information in the data format defined by *type*. Multiple *string* values can be coded for a *subfield_id type* pair. For example:

```
DLCADDR=(1,TYPE1,STRING1,STRING2,STRING3)
```

You can specify up to 250 bytes of signaling information on a single DLCADDR operand. If coding multiple *strings* on a single operand, the total number of bytes available for signaling information is reduced by 2 for each additional DLCADDR *string*.

For DLUR PUs, *string* defines either the DLC type or the DLC signaling information in the data format defined by *type*.

Multiple sets of DLCADDR operands can be coded on the PATH definition statement. The first DLCADDR operand for each set of DLCADDR operands must have a *subfield_id* of 1 and it identifies the DLC type, which can be one of the following:

- ATM
- Data link switching
- Ethernet
- FDDI
- Frame relay
- Internal PU
- SDLC leased
- SDLC switched
- Token-ring
- X.25

The remaining DLCADDR operands for each set of DLCADDR operands define the DLC signaling information for each DLC type. Refer to the data link control provider documentation (such as the *NCP Resource Definition Reference*) for information about how to code the remaining DLCADDR operands. The data link control provider is responsible for verifying the contents of the DLCADDR operand.

The maximum number of PATH statements in a switched major node that can have DLCADDR coded is 32767.

Note: DLCADDR is mutually exclusive with the LINENM operand for any single switched PATH definition statement.

Code DLCADDR, or DIALNO, or both for switched lines that can call out to a remote physical unit (that is, for links specified with CALL=OUT or CALL=INOUT on the LINE definition statement).

If both DLCADDR and DIALNO are coded on the PATH definition statement, both values are processed and stored. If the device that performs the routing of the data link connection supports expanded dial information, DLCADDR is used to make the connection. If the device does not support expanded dial information, DIALNO is used. Refer to the data link control provider documentation (such as the *NCP Resource Definition Reference*) for information about whether expanded dial information is supported, whether both DIALNO and DLCADDR are supported together, and how to code the DLCADDR contents. The data link control provider is responsible for verifying the contents of the DLCADDR operand.

For DLUR physical units: Code DLCADDR for switched connections where the DLUS host can call out to a remote physical unit that is supported by a DLUR. The information that is coded on the DLCADDR operand is dictated by the DLUR being used. Refer to the DLUR implementer documentation (README file) or to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how to code the DLCADDR contents. The DLUR implementer is responsible for verifying the contents of the DLCADDR operand.

For ATM native connections: Code the following DLCADDR operands for an ATM native connection. Fields in italics are user-defined fields.

Note: You can code only one set of DLCADDR operands on the PATH definition statement for ATM native connections.

```
DLCADDR=(1,C,ATMSVC,tg_use)
DLCADDR=(7,BCD,format,best_effort_indicator,
        a,b,c,forward_traffic_management_options,
        d,e,f,backward_traffic_management_options)
DLCADDR=(8,X,format,outbound_QOS_class,inbound_QOS_class)
DLCADDR=(9,BCD,tns_identifier)
DLCADDR=(21,X,type,remote_ATM_address,remote_ATM_address)
DLCADDR=(31,D,remote_service_access_point_address)
DLCADDR=(51,X,format,a,b,c,d,e)
DLCADDR=(61,BCD,format,05,a,b)
```

The following explains the information provided by each subfield:

Subfield
Contents

1 `C` indicates EBCDIC format. `ATMSVC` indicates that the connection provides native access to an ATM network.

Code *tg_use* to specify one of the following:

EXCLUSIVE
Indicates that the switched virtual channel (SVC) can be used only for this TG

SHARE
Indicates that the SVC can be used for this TG and connection network TGs from this port to the same remote node through the same remote ATM address

7 `BCD` indicates binary coded decimal format.

Code *format* to specify one of the following:
00 Indicates ITU-T defined
03 Indicates ATM network defined

Code *best_effort_indicator* to specify one of the following:
00 Indicates that guaranteed bandwidth is desired
01 Indicates that best effort is acceptable

Code *a*, *b*, *c*, *d*, *e*, and *f* as 4-byte decimal values to specify the following:

a Forward peak cell rate (cells per second) for cell loss priority 0+1 (CLP=0+1)

b Forward sustainable cell rate (cells per second) for (CLP=0)

c Forward maximum burst size (cells) for (CLP=0)

d Backward peak cell rate (cells per second) for cell loss priority 0+1 (CLP=0+1)

e Backward sustainable cell rate (cells per second) for (CLP=0)

f Backward maximum burst size (cells) for (CLP=0)

Code *forward traffic_management_options* to specify one of the following:

00 Indicates that tagging is not requested in the forward direction

01 Indicates that tagging is requested in the forward direction

Code *backward traffic_management_options* to specify one of the following:

00 Indicates that tagging is not requested in the backward direction

01 Indicates that tagging is requested in the backward direction

Note: If *best_effort_indicator* is 00, code *format*, *a*, *b*, *c*, *forward_traffic_management_options*, *d*, *e*, *f*, and *backward_traffic_management_options*.

If *best_effort_indicator* is 01, code *format*, *a*, and *d*.

8 `X` indicates hexadecimal format.

Code *format* to specify one of the following:
00 Indicates ITU-T defined
03 Indicates ATM network defined

Code *outbound_QOS_class* to specify the QOS class for outbound (forward) data traffic.

Code *inbound_QOS_class* to specify the QOS class for inbound (backward) data traffic.

9 `BCD` indicates bincary coded decimal format.

Code *tns_identifier* to specify the carrier identification code for the carrier that will be used for this connection.

21 `X` indicates hexadecimal format.

Code *type* to specify one of the following:

0101 Indicates public E164 address, which means the address is in a public ATM network. The address is variable in length.

0002 Indicates International Organization for Standardization (ISO) network service access point (NSAP), which means the address is in a private ATM network. The address is always 20 bytes long.

Code *remote_ATM_address* to specify the address of the remote node connected through this SVC.

Note: If the remote node is a VTAM node, refer to *OSA Planning* for information on how to determine the ATM address through which the remote VTAM node can be reached. You can also determine the ATM address by starting both VTAMs, then activating the following resources in each VTAM:
- The TRL major node
- The XCA major node that defines the port through which VTAM can be reached
- Any line defined in the XCA major node

Then issue a DISPLAY ID command for the XCA major node in each VTAM. The resulting displays show the ATM addresses for each VTAM. These are the addresses that should be specified in the switched major nodes of the VTAMs.

Code a second *remote_ATM_address* if *type*=0002 and the port through which this call is made is attached directly to a public network. This second *remote_ATM_address* is a public gateway E164 address through which the private address can be reached.

31 `D` indicates decimal format.

Code *remote_service_access_point_address* to specify the service access point address of the remote node connected through this SVC.

51 `X` indicates hexadecimal format.

Code *format* to specify the following:
00 Indicates ITU-T defined

Code *a*, *b*, *c*, *d*, and *e* to specify the following bearer capabilities:

a Bearer class.

a can be one of the following:
01 Indicates class A
03 Indicates class C
10 Indicates class X

b Susceptibility to clipping.

b can be one of the following:
00 Not susceptible
01 Susceptible

c User plane connection configuration.

c must be:
00 Point-to-point

d If bearer class is 10, indicates the traffic type.

d can be one of the following:
00 No indication
02 Variable bit rate

e If bearer class is 10, indicates timing requirements.

e can be one of the following:
00 No indication
01 End-to-end timing required

61 `BCD` indicates binary coded decimal format.

Code *format* to specify the following:
00 Indicates ITU-T defined

Code 05 to indicate ATM adaptation layer (AAL) type 5.

Code *a* and *b* as 4-byte decimal values to specify the following AAL parameters:
a Forward CPCS-SDU maximum size
b Backward CPCS-SDU maximum size

Note: If an application using an ATM reserved bandwidth SVC connection sends RUs smaller than the forward CPCS-SDU size, the throughput for the session may be reduced. For example, if the forward CPCS-SDU size was defined or allowed to default to 8448 (Maximum RU size 8K + 256), and the data being transmitted over this ATM connection is in 1K RUs, the throughput (as defined by the connection's forward cell rate) is greatly reduced. If this ATM SVC connection is used for 1K RU data transmission, the forward CPCS-SDU size should be set to 1280 bytes (Maximum RU size of 1K + 256).

DLOGMOD

```
►►──┬──────────────────────────────────────────────────────────────────────────►◄
    └─DLOGMOD=default_logon_mode_table_entry_name─┘
```

statements: *GROUP, PU, LU*
dependencies: *PUTYPE=1 or 2*

Specifies the name of the logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used by default).

If you specify MODETAB, the entry must be in either the specified table or in ISTINCLM, an IBM-supplied default logon mode table. If you do not specify MODETAB, the entry must be in ISTINCLM. For more information on logon mode entries see “Logon Mode Table” on page 698.

DLURNAME

```
►►──┬──────────────────────────────────────────────────────────────────────────►◄
    └─DLURNAME=dlur_name─┘
```

statements: *PATH*
dependencies: *DLCADDR*

Specifies the name of the dependent LU requester (DLUR) for this PU. You can code a network-qualified name for *dlur_name*.

If you do not code NETID, the network ID for the DLUR defaults to the network ID of the dependent LU server (DLUS). Note that NETID specifies a real network ID and that an alias network ID is not allowed for the DLUR.

The data link control (DLC) information specified on the DLCADDR operand is passed to the DLUR specified.

Note: If DLURNAME is specified, do not code the following operands:
CALL
GID
GRPNM
REDIAL
SHMTIM
VERID
VERIFY

Note that the redial function is supported automatically for DLUR PUs, and the REDIAL operand is not involved. However, the following restrictions apply to the redial function for DLUR PUs:

- Redial applies only to PUs activated by dependent LU servers.
- Redial is attempted as many times as there are PATH definition statements for a PU.

- Redial is attempted for any sense code except 082C0002 (PU is already active) or 083B0002 (PCID is not unique). For 083B0002, VTAM deletes the older PCID and then resends the activation request with a new PCID.
- Redial is attempted for every active or pending active PU served by a DLUR when a protocol violation, topology database update error, or session outage signal is received for that DLUR.

DWACT

```
►►─┬─DWACT=NO──────────┬──────────────────────────────────────────►◄
   │                   │
   └─DWACT=─┬─NO──┬────┘
            └─YES─┘
```

statements:	*PU*
dependencies:	*none*

Specifies whether a physical unit is dialed when it is activated. DWACT provides an alternative to using the VARY DIAL command on each physical unit after it is activated.

For more information on the use of DWACT with a switched physical PU which has an associated logical hierarchy, see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

DWACT=NO
Specifies that the physical unit is not dialed after it is activated.

DWACT=YES
Specifies that the physical unit is dialed after it is activated.

Note: DWACT=YES has no effect on VARY INACT,R command processing; VARY INACT,R causes the specified resource to be inactivated and reactivated but no dial operation will be performed.

DYNADJCP

```
►►─┬─DYNADJCP=value_of_dynadjcp_start_option──────────┬──────────────────────────────────►◄
   │                                                  │
   │                 ┌─value_of_dynadjcp_start_option─┐
   └─DYNADJCP=───────┼─NO────────────────────────────┼──┘
                     └─YES───────────────────────────┘
   ┌─DYNADJCP=YES──┐  (1)
   │               │
   └─DYNADJCP=YES──┘
```

Notes:

1 DYNADJCP=YES is the only valid value if the TRLE operand is coded.

statements:	*GROUP, PU*
dependencies:	*APPN only; ignored if CPNAME coded*

Specifies whether an adjacent control point can be created dynamically for the adjacent node.

DYNADJCP=NO
Indicates that an adjacent control point cannot be dynamically created for the adjacent node.

DYNADJCP=YES
Indicates that an adjacent control point can be dynamically created for the adjacent node.

Note: A type 2.1 peripheral node in the switched major node requires both IDBLK and IDNUM, or CPNAME. DYNADJCP=YES (the operand or the start option) is valid when you code IDBLK and IDNUM. However, DYNADJCP is ignored when you code CPNAME.

DYNLU

```
      ┌─DYNLU=value_of_dynlu_start_option──────┐
►►────┼──────────────────────────────────────────┼──────────────────────────────────►◄
      │          ┌─value_of_dynlu_start_option─┐│
      └─DYNLU=───┼─────────────────────────────┼┘
                 ├─NO──────────────────────────┤
                 └─YES─────────────────────────┘
      ┌─DYNLU=YES─┐  (1)
      │           │
      └─DYNLU=YES─┘
```

Notes:

1 DYNLU=YES is the only valid value if the TRLE operand is coded.

statements:	*GROUP, PU*
dependencies:	*PUTYPE=2; CDRDYN*

Specifies whether dynamic allocation of CDRSC definitions is supported. If you do not code this operand, the default is the value you coded on the DYNLU start option.

If dynamic allocation of CDRSC definitions is not supported for a particular PU, a resource that was dynamically defined as a CDRSC cannot use that PU to connect to a session.

This operand is valid only when PUTYPE=2, and it does not apply to SSCP-SSCP CDRSCs.

DYNLU=NO
Specifies that dynamic allocation of CDRSC definitions is not supported. If you code DYNLU=NO and plan to connect to other nodes using CP-CP sessions, then predefine each node in the CDRSC major node. If you do not, CP-CP session setup fails.

DYNLU=YES
Specifies that dynamic allocation of CDRSC definitions is supported.

For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

For APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful even though a host CDRM cannot be activated.

Note: If you have coded a DYNLU value in the adjacent control point major node for the adjacent control point to which this physical unit is linked, that value overrides the DYNLU value you code on this PU definition statement. For additional DYNLU considerations, see “DYNLU” on page 21.

ENCR

```
►►─┬─ENCR=NONE──────────┬──────────────────────────────────────────►◄
   │                    │
   └─ENCR=─┬─NONE─┬─────┘
           ├─OPT──┤
           ├─REQD─┤
           └─SEL──┘
```

statements: GROUP, PU, LU
dependencies: none

Indicates whether this logical unit has any special requirements for enciphering and deciphering messages.

ENCR=NONE
Specifies that this logical unit cannot engage in cryptographic sessions.

ENCR=OPT
Specifies that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

ENCR=REQD
Specifies that VTAM must encipher all messages to and from this logical unit. If ENCR=REQD, no sessions can be established with this logical unit unless the host with which it is associated can handle cryptographic sessions.

ENCR=SEL
Has no meaning for logical units. If you specify ENCR=SEL, ENCR=OPT is used instead.

ENCRTYPE

```
►►─┬─ENCRTYPE=DES────────┬──────────────────────────────────────────►◄
   │                     │
   └─ENCRTYPE=TDES24─────┘
```

statements: *GROUP, PU, LU*
dependencies: *none*

Specifies the minimum type of encryption that VTAM should use on behalf of the application when performing session level encryption.

ENCRTYPE=DES
Specifies that VTAM must use a minimum of DES encryption with an 8–byte key when performing session level encryption. This is the default.

ENCRTYPE=TDES24
Specifies that VTAM must use a minimum of Triple_DES encryption with a 24–byte key when performing session level encryption.

Note: The ENCRTYPE value coded on the GROUP or PU will be sifted down to the appropriate resource (LU).

The SYSDEFed ENCRTYPE of a LU can be changed at runtime through the MODIFY SECURITY command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for specific details. However, the ENCRTYPE value will not be able to be downgraded. In other words, if the application is currently operating at ENCRTYPE=TDES24, then the encryption level cannot be changed to ENCRTYPE=DES using the MODIFY SECURITY command.

Note: If downgrading the security level is preferred, the current major node being used must be deactivated and another major node value must be coded.

In addition, a user may use the LOGMODE TABLE to override the ENCRTYPE value of the SLU based on the logmode entry being used. The ENCRTYPE value will not be able to be downgraded from TDES24 to DES using the logmode table. Only TDES24 can be specified for the encryption level (ENCRTYP). Refer to “MODEENT Macroinstruction” on page 699 for more information on MODEENT macroinstructions.

FASTPASS

```
►►─┬─FASTPASS=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─FASTPASS=─┬─YES─┬─────┘
               └─NO──┘
```

statements:	*GROUP, PU, LU*
dependencies:	*TSO sessions only*

Determines how session establishment is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure. The application program acts as a secondary logical unit (SLU).

The PLU acknowledges the capability of the SLU during session establishment. If you have coded the logon performance enhancement in the PLU’s application program, only a single bind is issued to establish the session.

Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

Note: FASTPASS applies only to application programs (acting as SLUs) that establish sessions with TSO.

FASTPASS=NO
Specifies that a pre-V3R3 logon procedure is performed for application programs that issue the CLSDST macroinstruction with the PASS as part of their session establishment procedure.

FASTPASS=YES
Specifies that the logon performance enhancement is performed for application programs that issue the CLSDST macroinstruction with the PASS option as part of their session establishment procedure.

FEATUR2

```
   ┌─FEATUR2=(DUALCSE,MODEL1,NOEDATS,NOPRINTR,NOSELPEN)─┐
►►─┤                                                    ├──────────────────────────►◄
   └─FEATUR2=(──────────────────────────────────────────)──────────────────────────
                 ┌─DUALCSE──┐
                 ├──────────┤
                 └─LOWERCSE─┘
                 ┌─MODEL1───┐
                 ├──────────┤
                 └─MODEL2───┘
                 ┌─NOEDATS──┐
                 ├──────────┤
                 └─EDATS────┘
                 ┌─NOPRINTR─┐
                 ├──────────┤
                 └─PRINTR───┘
                 ┌─NOSELPEN─┐
                 ├──────────┤
                 └─SELPEN───┘
```

statements:	*GROUP, PU, LU*
dependencies:	*none*

Contains certain device information for virtual logical units that are supported by network terminal option (NTO).

FEATUR2=EDATS

FEATUR2=NOEDATS
Specifies whether this terminal has the extended data stream feature. You cannot use this operand for terminals attached by SDLC lines.

FEATUR2=DUALCSE

FEATUR2=LOWERCSE
Specifies how VTAM sends alphabetical characters coded with the TEXT operand on a USSMSG macroinstruction to a non-SNA terminal over the SSCP-LU session. This value does not affect non-alphabetical characters, or any characters coded on the BUFFER operand of a USSMSG macroinstruction.

Code LOWERCSE to indicate that alphabetical characters are sent to the terminal over the SSCP-LU session in lowercase. Code DUALCSE to indicate that VTAM sends all characters as they are coded in the USSMSG macroinstruction.

FEATUR2=MODEL1

FEATUR2=MODEL2
Identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

This information is available to an application program as part of the device characteristics pertaining to this terminal. You can obtain those characteristics by using the INQUIRE macroinstruction. For more information on using the INQUIRE macroinstruction, see *OS/390 eNetwork Communications Server: SNA Programming*.

FEATUR2=NOPRINTR

FEATUR2=PRINTR
Specifies whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only if TERM=3275.

FEATUR2=NOSELPEN

FEATUR2=SELPEN
Specifies whether this terminal supports a selector pen.

GID

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └GID=group_identifier──┘
```

statements:	*PATH*
dependencies:	*none*
range:	*1–255*

Specifies the group identifier of a group of paths across all physical units in the switched major node.

Group identifiers can be assigned to allow the VTAM operator to regulate the use of switched network services. For example, if GID=6 is assigned to all paths in a switched major node that use direct distance dialing, the VTAM operator can make all of the paths usable or not usable with a single command.

GRPNM

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └GRPNM=group_name──────┘
```

statements:	*PATH*
dependencies:	*CALL; VERID; VERIFY*

Provides the name of a GROUP definition statement in the NCP or channel-attachment major node that defines a group of SDLC switched links. The line group must have all the characteristics necessary to process the telephone number and must be compatible with the type of physical unit.

GRPNM is required when one of the following is specified:
- VERIFY=IN, OUT, or INOUT
- VERID
- CALL

A dial-out connection will be made to the PU associated with this PATH definition statement.

Note: When defining an NCP/Token-Ring Interconnection (NTRI), an IBM 3172 Nways Interconnect Controller, an IBM S/390 Open Systems Adapter group, *group_name* should be the name of the logical group definition, or an IP network for Enterprise Extender.

HOSTNAME

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─HOSTNAME=remote_host_name─┘
```

statements: *PATH*
dependencies: *MEDIUM=HPRIP, HOSTNAME is ignored if IPADDR is coded*

Specifies the hostname used by the TCP/IP stack on the remote end of the Enterprise Extender connection. The HOSTNAME you code may be qualified with its domain name. The local host must be able to perform TCP/IP name-to-address resolution, either by connectivity to a DNS name server, or by using a local host, HOSTS.SITEINFO. For more information refer to the OS/390 SecureWay Communications Server: IP Configuration. Furthermore, if the hostname represents a CS for OS/390 APPN node, the address returned must be a VIPA address.

Notes:

1. Remote_host_name may be up to 255 characters in length.
2. HOSTNAME and IPADDR are mutually exclusive, however, IPADDR is used if both are specified.
3. HPR must equal YES if the PU is used for Enterprise Extender.

HPR

```
    ┌─HPR=YES──────┐
►►──┤              ├──────────────────────────────────────────────────────►◄
    └─HPR=──┬─NO──┬─┘
            └─YES─┘
```

statements: *GROUP, PU*
dependencies: *APPN only*

Specifies whether high performance routing (HPR) support is enabled for a PU.

HPR=YES
Indicates that this PU provides HPR support.

HPR=NO
Indicates that this PU does not provide HPR support.

IDBLK

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └─IDBLK=id_block────────┘
```

statements: *PU*
dependencies: *CPNAME; IDNUM*

range: *X'001'–X'FFE'*

Specifies a 3-digit hexadecimal number that identifies the device type. You can get the number from the component description manual for the device. All devices of the same type, such as all 3274s, will have the same IDBLK.

Valid values are X'001'–X'FFE'. Note that if the value is X'nnn', you code IDBLK=nnn. The IDBLK number occupies bits 16–27 of the 48-bit station ID constructed by VTAM for switched network operation. See the IDNUM operand for a description of the station ID.

You should not code IDBLK for switched subarea connections (type 4 and 5 physical units).

For any switched peripheral node (PUTYPE=1 or 2):

- If the node is not a type 2.1 peripheral node, you must code both IDBLK and IDNUM.
- For a type 2.1 peripheral node, coding CPNAME is required if the peripheral node supports role negotiation (such as 3172, 3174, and Communications Manager).

  For external communication adapter type 2.1 peripheral nodes, you must code CPNAME. IDBLK and IDNUM are ignored if you code them.

  For an APPN connection, if you code IDBLK and IDNUM, code DYNADJCP=YES instead of CPNAME.
- If VTAM is defined as a type 2.1 node, you must code CPNAME.
- See the appropriate product manual for CPNAME and IDNUM and IDBLK values.

IDNUM

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─IDNUM=id_number──┘
```

statements: *PU*
dependencies: *CPNAME; IDBLK; PUTYPE=4,5*

Specifies a 5-digit hexadecimal number that identifies the specific device (if PUTYPE=1 or 2) or the specific connection (if PUTYPE=4 or 5). The IDNUM for the device is usually the serial number of the device. The device's serial number is frequently used for this purpose. You can get this number from the component description manual for the device.

For any switched peripheral node (PUTYPE=1 or 2):

- If the node is not a type 2.1 peripheral node, you must code both IDBLK and IDNUM.
- For a type 2.1 peripheral node, coding CPNAME is required if the peripheral node supports role negotiation (such as 3172, 3174, and Communications Manager). For external communication adapter type 2.1 peripheral nodes, you must code CPNAME. IDBLK and IDNUM are ignored if you code them. For an APPN connection, if you code IDBLK and IDNUM, code DYNADJCP=YES instead of CPNAME.
- If VTAM is defined as a type 2.1 node, code CPNAME.
- See the appropriate product manual for CPNAME and IDBLK and IDNUM values.

For peripheral physical units, VTAM uses IDBLK and IDNUM to build a 48-bit station ID that is used in XID exchange during the dial procedure. That station ID must be unique for each station within the network, not only within the major node.

The station ID is structured as follows:

Bits	Meaning
0–3	Reserved
4–7	PUTYPE
8–15	X'00'
16–27	IDBLK
28–47	IDNUM

For subarea physical units (PUTYPE=4 or 5), code either IDNUM or CPNAME. VTAM uses the IDNUM value to correlate the PU definitions between the two SSCPs that own each end of the switched line. This IDNUM value must be unique for all PU definitions within this SSCP that contain the same subarea number and NETID.

Note: Specify the same IDNUM on the PU definitions in the two SSCPs that own each end of the switched lines.

The node identifier field of an XID contains IDNUM along with a value of X'FFF' as the IDBLK.

For subarea physical units, VTAM uses the subarea number specified on SUBAREA and the IDNUM value to build a station ID that is used in some operator messages. That station ID must be unique for each station within the network, not only within the major node.

The station ID is structured as follows:

Bits	Meaning
0–15	Subarea number
16–27	X'FFF'
28–47	IDNUM

INCLUD0E

```
     ┌─INCLUD0E=NO──────┐
►►───┤                  ├──────────────────────────────────────────────────►◄
     └─INCLUD0E=──┬─NO──┬─┘
                  └─YES─┘
```

statements: *PU*
dependencies: *none*

Specifies whether VTAM provides LU names to the logical units defined under this PU. The names are provided by passing a CV0E, Network Name control vector on the ACTLU requests.

INCLUD0E=NO
Indicates that VTAM will not provide LU names to the logical units defined under this PU.

INCLUD0E=YES
Indicates that VTAM will provide LU names to the logical units defined under this PU.

IPADDR

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └IPADDR=address────┘
```

statements:	*PATH*
dependencies:	*MEDIUM=HPRIP*

Specifies the remote TCP/IP address for the connection.

Notes:

1. The address must be specified in dotted decimal format (for example, IPADDR=4.6.41.7).
2. HOSTNAME and IPADDR are mutually exclusive. However, IPADDR is used if both are specified.

IPRESOLV

```
    ┌─IPRESOLV=(45)────┐
►►──┤                  ├──────────────────────────────────────────►◄
    └─IPRESOLV=(1-255)─┘
```

Specifies the number of seconds VTAM waits for IP address resolution request to complete before canceling the request. The default is 45 seconds. Any value from 1 to 255 seconds may be specified.

IRETRY

```
    ┌─IRETRY=NO──────────┐
►►──┤                    ├──────────────────────────────────────────►◄
    │            ┌─NO──┐ │
    └─IRETRY=────┤     ├─┘
                 └─YES─┘
```

statements:	*GROUP, PU*
dependencies:	*PUTYPE=1 or 2*

Specifies whether the boundary NCP (the NCP to which the switched physical unit becomes connected) retries a polling operation immediately for the device if an idle detect timeout follows a polling operation. For more information on this operand, see the *NCP, SSP, and EP Resource Definition Reference*.

IRETRY=NO
Specifies that the boundary NCP does not retry the polling operation.

IRETRY=YES
Specifies that the boundary NCP retries the polling operation.

ISTATUS

```
►►─┬─ISTATUS=ACTIVE──────────┬──────────────────────────────────────►◄
   │                         │
   └─ISTATUS=─┬─ACTIVE───┬───┘
              └─INACTIVE─┘
```

statements: *GROUP, PU, LU*
dependencies: *none*

Determines whether the resource is to be activated after the first activation of the switched major node.

ISTATUS=ACTIVE
Indicates that the resource becomes active when its major node becomes active.

ISTATUS=INACTIVE
Indicates the resource is to remain inactive until the operator activates it. The operator can override this value by specifying the SCOPE operand on the VARY ACT command when the major node is activated. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the VARY ACT command.

LIMRES

```
      ┌─LIMRES=NO──────┐
►►────┼────────────────┼──────────────────────────────────────────►◄
      │                │
      └─LIMRES=─┬─NO──┬┘
                └─YES─┘
      ┌─LIMRES=NO─┐  (1)
      │           │
      └─LIMRES=NO─┘
```

Notes:

1 LIMRES=NO is the only valid value if the TRLE operand is coded.

statements: *GROUP, PU*
dependencies: *LIMQSINT; LU 6.2 only; PUTYPE=1 or 2; Switched dynamics*

Specifies whether the physical unit is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes any sessions that traverse the resource to be deactivated if no conversation is active. A session can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

If you do not code a value for LIMRES, resources are not considered as limited resources. Furthermore, sessions will remain active even after all conversations have ended.

You can also use LIMRES in conjunction with the LIMINTCP start option to specify the interval to retain a free session using a limited resource for some network management flows (excluding flows using the APPN control point sessions or dependent LU server sessions). If you do not code a value for the LIMINTCP start option, resources are not considered to be limited resources, and the sessions will remain active even after all conversations have ended.

LIMRES=NO
Specifies that this physical unit is not to be treated as a limited resource.

LIMRES=YES
Specifies that this physical unit is to be treated as a limited resource.

LINENM

```
►►──LINENM=line_name──────────────────────────────────────────────────►◄
```

statements:	*PATH*
dependencies:	*DIALNO; DLCADDR*

Names a line for the direct call function.

LINENM is mutually exclusive with the DIALNO and DLCADDR operands for any single switched path definition statement.

Although not required, the PATH definition statement for a direct call line should be the first statement following the switched PU definition statement. This ensures that VTAM attempts to call the switched physical unit using the direct call line first.

Note: The direct call function is not available for switched subarea connections.

LLERP

```
      ┌─LLERP=NOTPREF──────┐
►►────┤                    ├──────────────────────────────────────────►◄
      └─LLERP=──┬─REQUIRED─┬─┘
                └─NOTPREF──┘
```

statements:	*PU*
dependencies:	*Type 2.1, HPR=YES*

Specifies the link-level error recovery procedures (LLERP) preference for HPR connections. The LLERP value will be overridden to NOTPREF for PUs in a switched major node that will be mapped to LSA (XCA major node).

LLERP=REQUIRED
Specifies that link-level error recovery procedures are required.

LLERP=NOTPREF
Specifies that link-level error recovery procedures are required by this PU only when the adjacent link station requires it.

LOCADDR

```
►►──LOCADDR=address──────────────────────────────────────────────────►◄
```

statements:	*LU*
dependencies:	*none*
range:	*0–255 (PU type 2) 0–63 (PU type 1)*

Specifies the logical unit's local address at the physical unit. For dependent logical units, the range of valid local addresses depends on the PUTYPE of the associated physical unit. All dependent logical units under a given physical unit must have unique LOCADDR values.

An LU definition statement is not required for every possible local address, and LOCADDR values need not be consecutive. Unused local addresses smaller than the largest local address at a station are not assigned network resources.

Notes:

1. The value of LOCADDR depends upon the requirements of the device being defined. Consult the appropriate component description manual for these restrictions.
2. Although you can use this operand to define independent LUs, it is recommended that you either use dynamic definition or predefine the independent LUs as CDRSCs.

LOGAPPL

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─LOGAPPL=controlling_primary_lu─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*

Identifies the name of the primary logical unit that this logical unit will automatically log on to as a secondary logical unit when this logical unit is activated. The name you specify can also be a USERVAR that will be resolved into its current value before being used to establish or redrive the session with the controlling primary logical unit.

For independent logical units, the LOGAPPL operand establishes a session with the primary logical unit designated on the LOGAPPL operand if a session does not currently exist with this logical unit. The session is established regardless of whether sessions with other logical units exist. The LOGAPPL operand cannot contain the name of an independent logical unit because independent logical units cannot be the controlling logical units for an automatic logon session.

To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start option.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Notes:

1.

If an error occurs for the switched connection (an INOP), VTAM reallocates the LOGAPPL session and attempts to redial the logical unit to recover the connection. The original session is ended by disconnecting the dial connection instead of ending the session normally using LOGOFF and TERMSELF.

If you do not code the LOGAPPL operand, the PU and LU must be activated manually.

2. If you specify the NetView or TSO program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LOGTAB

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └LOGTAB=interpret_table_name┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*

Specifies the name of an interpret table that VTAM uses when processing logon requests originating from the logical unit. VTAM uses the interpret table to interpret the name received from an initiation request. See “Interpret Table” on page 690 for a description of how to define an interpret table.

LOGTAB is valid for dependent logical units only.

LUGROUP

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └LUGROUP=lu_group_name┘
```

statements:	*PU*
dependencies:	*none*

Specifies the name of the model LU group that VTAM uses to select a model LU definition when dynamically defining an LU. This operand also indicates that the device supports dynamic definition of LUs.

LUSEED

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └LUSEED=lu_pattern_name┘
```

statements:	*PU*
dependencies:	*LUGROUP*

Provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- The first character must be either alphabetical or one of the national characters (@ or $).
- The remaining characters can be either alphabetical, numerical, or national.
- There must be either two or three contiguous national # characters in the name. These characters represent holders, in which the logical unit's local address is substituted when an LU's name is generated. If you specify three # characters, the LU's logical address is substituted in decimal for the # characters. If you specify two # characters, the LU's logical address is substituted in hexadecimal for the # characters.

When used with a user-written SDDLU exit routine, this operand is optional and there are no restrictions on its format. For more information on the SDDLU exit, see *OS/390 SecureWay Communications Server: SNA Customization*.

This operand is valid only when you code the LUGROUP operand.

MAXDATA

For PU type 1:

```
►►─┬─MAXDATA=261──────────────────┬──────────►◄
   └─MAXDATA=number_of_bytes──────┘
```

statements:	GROUP, PU
dependency:	see notes
range:	5-65535

For PU type 2:

```
►►─┬─MAXDATA=265──────────────────┬──────────►◄
   └─MAXDATA=number_of_bytes──────┘
```

statements:	PU
dependency:	see notes
range:	5-65535

Specifies the maximum number of bytes that the physical unit can receive in one PIU or PIU segment, including the transmission header (TH) and the request or response header (RH). To determine the maximum PIU (or PIU segment) size that the physical unit can receive, consult the component publications for the specific type of SDLC station represented by this PU definition statement.

The maximum amount of user data that VTAM sends to the physical unit in one PIU is:

- MAXDATA minus 2 bytes (for a PU type 1)
- MAXDATA minus 6 bytes (for a PU type 2)

Note: The RU part of the PIU contains the user data. When VTAM segments the PIU, the middle and last PIUs will contain three more bytes of user data in the RU because there is no RH field. The value for the PIU varies with the device; for example, the 3276 requires MAXDATA=262.

You must specify a MAXDATA value for a type 4 or 5 physical unit; there is no default for these physical units.

Notes:

1. In a local area network, the value for the MAXDATA operand on the PU definition statement must be less than or equal to the value for the MAXDATA operand on the PORT definition statement for the LAN major node to which the switched physical unit is connected.

2. NCP V4R3 changed the MAXDATA calculation for a type 1 physical unit. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information on calculating a MAXDATA value for physical units on nonswitched links.
3. MAXDATA is ignored for type 2.1 nodes attached through an NCP, an IBM 3172 Nways Interconnect Controller, or through an IP network for Enterprise Extender. The type 2.1 node supplies the actual value used when the connection is established.

MAXDLUR

```
►►──┬──────────────────────────────────────────┬──────────────────────────────────►◄
    └─MAXDLUR=max_number_of_unique_DLURs─────────┘
```

statements:	*VBUILD*
dependencies:	*valid for NN and ICN only; DLURNAME coded*
range:	*1–32766*

Specifies the maximum number of dependent LU requesters. MAXDLUR should be coded only when DLURNAME is coded.

MAXGRP

```
►►──┬──────────────────────────────────────────┬──────────────────────────────────►◄
    └─MAXGRP=max_number_of_unique_path_groups───┘
```

statements:	*VBUILD*
dependencies:	*PATH*
range:	*0–32766*

Specifies the number of unique path groups (group names) that are defined in the GRPNM operand of all PATH definition statements within the switched major node. Code MAXGRP only if you also code the switched PATH definition statement.

MAXNO

```
►►──┬──────────────────────────────────────────┬──────────────────────────────────►◄
    └─MAXNO=max_number_of_unique_dial_digits────┘
```

statements:	*VBUILD*
dependencies:	*PATH*
range:	*0–32766*

Specifies the number of unique telephone numbers that are defined in the DIALNO operand of all PATH definition statements within the switched major node. Code MAXNO only if you also code the switched PATH definition statement.

For VTAM LAN support, if you code switched PATH definition statements with no DIALNO operands, you must code a value of 1 for the MAXNO operand.

MAXOUT

```
►►─┬─MAXOUT=1──────────────────┬──────────────────────────────────────────►◄
   └─MAXOUT=max_number_of_pius─┘
```

statements:	*GROUP, PU*
dependencies:	*ignored for type 2.1 if attached through an IBM 3172 Nways Interconnect Controller*
range:	*1–127*

Specifies the maximum number of PIUs or PIU segments (if the session parameters allow segmenting of data) that VTAM sends to the physical unit represented by this definition statement before requesting a response.

For performance reasons, you should code MAXOUT=7 if the physical unit represented by this definition statement can handle more than one PIU before sending out a link-level response. To determine this, consult the component publications for the specific type of SDLC station.

Note: MAXOUT is ignored for type 2.1 nodes attached through an IBM 3172 Nways Interconnect Controller or through an IP network for Enterprise Extender.

MAXSESS

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MAXSESS=max_number_LU-LU_sessions─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*independent LUs attached to an NCP*
range:	*1–65535*

Specifies the maximum number of concurrent LU-LU sessions in which an independent LU can participate. The MAXSESS value applies to each connection the LU is using to establish LU-LU sessions. For example, if MAXSESS=2 was specified for LUA and LUA appears on three PUs, LUA can have two sessions on each of the three PUs for a total of six sessions.

If you do not code MAXSESS, or code a value outside the range, the MAXSESS value set on the BUILD definition statement (NCP major node) will be used. Reference the *NCP, SSP, and EP Resource Definition Reference* for more information about this operand.

MDLENT

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └─MDLENT=model_name_table_entry_name─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*MDLTAB; PUTYPE=1 or 2*

Specifies the name of the model name table entry to be used for this logical unit.

If you code the MDLENT operand, you must also code the MDLTAB operand. If you omit MDLENT and include MDLTAB, VTAM uses the first entry in the table named in MDLTAB by default.

MDLTAB

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─MDLTAB=model_name_table_name─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*

Specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 720 for a description of how to define a model name table.

The model name is only passed when VTAM initiates the session with the application program, for example, only when the LOGAPPL = parameter is also coded.

If you omit the LOGAPPL operand or the MDLTAB operand, VTAM will not provide the name of the model name table to the application during LU-LU session initiation. However, you can provide the model name in other ways. You can enter it from the terminal on the MODEL operand of the LOGON command, or VTAM can send it to the application in a formatted session-initiation request. See “LOGON Command” on page 758 for more information on the MODEL operand of the LOGON command.

You can use the MODIFY TABLE command to dynamically replace a model name table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about the MODIFY TABLE command.

MLTGPRI

```
    ┌─MLTGPRI=0────────┐
►►──┤                  ├──────────────────────────────────────────────►◄
    └─MLTGPRI=priority─┘
```

statements:	*PU*
dependencies:	*none*

Assigns a priority to a line in a MLTG. The lowest priority is 0 and the highest priority is 255. NCP selects higher priority lines before lower priority lines when you code MLTGORDR=MLTGPRI on the BUILD definition statement. If you code MLTGORDR=FIFO on the BUILD definition statement or if you do not code MLTGORDR, the MLTGPRI keyword is ignored.

MODETAB

```
►►─┬─MODETAB=ISTINCLM──────────────────┬─────────────────────────────────────►◄
   │                                   │
   └─MODETAB=logon_mode_table_name─────┘
```

statements: *GROUP, PU, LU*
dependencies: *PUTYPE=1 or 2*

Specifies the name of a logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 698. If you do not supply a logon mode table for the logical unit on the MODETAB operand, an IBM-supplied default logon mode table (ISTINCLM) is used. If you specify a table, both the table you specify and the default table are used.

name

```
►►─┬──────────┬─────────────────────────────►◄   ►►─name──────────────────────►◄
   └─name─────┘
```

statements: VBUILD, PATH *statements:* GROUP, PU, LU

Specifies the name associated with the resource being defined.

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node’s status.

On PATH: Provides an optional name for the PATH statement.

On PU: Provides the required minor node name of the physical unit represented by this definition statement.

For an APPN PU, *name* must differ from *control_point_name* (specified on CPNAME) to ensure that a CP-CP session establishment succeeds. If *name* and *control_point_name* are duplicates, the session establishment fails and sense code 08880001 is given.

On LU: Provides the required minor node name of the logical unit represented by this definition statement.

NATIVE

```
►►─┬──────────────────────────┬─────────────────────────────────────────────►◄
   └─NATIVE=─┬─NO──┬──────────┘
             └─YES─┘
```

statements: *GROUP, PU*
dependencies: *BN=YES; APPN only*

Specifies whether this link station represents a connection to a native node.

If NATIVE is not specified on the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

No subnetwork negotiations will occur if NATIVE has been specified; in this case the value between the two nodes must match for connection to occur.

If you code NATIVE on both the ADJCP and PU definition statements, the values must match.

For transmission groups (TGs) between nodes, the value specified or negotiated for NATIVE must match. If a TG is already active, subsequent TG activations must not have a specified or negotiated value for NATIVE that conflicts with the value of NATIVE for the active TG. If there is a conflict, the TG attempting activation is not allowed to become active. If the activation is for a virtual-route-based transmission group then the SSCP-SSCP session will also fail.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─NETID=network_id─┘
```

statements: *PU*
dependencies: *XNETALS start option*

Specifies a 1–8 character network identifier.

Note: The use of the national character, #, is discouraged, because it might not be on keyboards of terminals produced in some countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

NETID=*network_id*

For type 4 and 5 physical units: NETID identifies the network in which this physical unit resides. It is required for a connection to another network.

If you do not specify a value, the network specified as NETID in the VTAM start options is used.

For type 2 physical units: This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See “Start Options” for information about XNETALS.

If you specify NETID on the PU definition statement, VTAM ensures that when the physical unit is active, the connecting resource is within the network

specified by NETID. If you omit NETID, VTAM dynamically processes the network ID when connections are established. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM's network id.

If a dynamic nonnative dial-out connection is required (that is, a session request drives the dial), and it is not an APPN connection, then NETID must be predefined. This ensures that subsequent session requests do not fail due to VTAM's inability to locate dependent logical units by using the dynamic nonnative NETID of the physical unit. This is also true when establishing a DLUS pipe, which is an APPN connection.

For dependent LU server PUs, VTAM will not dynamically process the network ID when CPSVRMGR sessions are established. To correctly define the NETID of a dependent LU server, if the NETID coded on the DLURNAME operand on the PATH statement is different from VTAM's network ID, the NETID parameter on the PU statement must be coded with the NETID coded on the DLURNAME otherwise the VTAM's network ID will be used for the PU.

NN

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─NN=──┬─NO──┬─────┘
           └─YES─┘
```

statements: *PU*
dependencies: *APPN only*

Specifies whether the adjacent node is expected to be a network node. If a value for the NN operand is specified, it will be validated when a connection is attempted to the adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If the NN operand is not specified, then the APPN capabilities of the adjacent node are identified and accepted when a connection is established.

NN=NO
Specifies that the adjacent node is expected to be an end node.

NN=YES
Specifies that the adjacent node is expected to be a network node.

PACING

```
    ┌─PACING=1──────────────┐
►►──┼───────────────────────┼──────────────────────────────────────►◄
    │                       │
    └─PACING=──┬─1──────────┬┘
               └─pacing_number─┘
```

statements: *GROUP, PU, LU*
dependencies: *PUTYPE=1 or 2*
range: *0–63*

Specifies how VTAM paces the flow of data from the boundary node, which performs pacing for a channel-attached SNA device, to the SLU. When the SLU and the PLU are in the same domain, the PACING value is ignored.

Note: Your can also use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and use fixed pacing instead.

To use the FIXED option, you must code (N,,FIXED). The second comma represents a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

PASSLIM

```
►►─┬─PASSLIM=1──────────────────┬──────────────────────────────────────────►◄
   │                            │
   └─PASSLIM=number_of_pius─────┘
```

statements:	*GROUP, PU*
dependencies:	*PUTYPE=1 or 2*
range:	*1–MAXOUT value*

Determines, for switched physical units connected through an NCP, the maximum number of contiguous PIUs that the NCP sends to the physical unit at one time.

If the switched PU is a type 4 or 5 PU, this operand is ignored. In a switched subarea environment, the value on PASSLIM is overridden with the value of MAXOUT.

PDELAY

```
   ┌─PDELAY=TERRESTR──────────┐
   │                   (1)    │
   ├─PDELAY=NEGLIGIB──────────┤
►►─┤                          ├──────────────────────────────────────────►◄
   │            ┌─TERRESTR─┐  │
   └─PDELAY=────┤          ├──┘
                ├─LONG─────┤
                ├─NEGLIGIB─┤
                └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

statements:	*GROUP, PU*
dependencies:	*APPN only*

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=NEGLIGIB
Indicates a local area network delay (less than .48 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between .48 and 49.152 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PID

```
►►──┬──────────────────────────┬──────────────────────────────────────►◄
    └─PID=path_identifier──────┘
```

statements:	*PATH*
dependencies:	*none*
range:	*1–255*

Specifies an identifier for the path being defined. The operator uses this identifier to change the status of the path.

This identifier is unique for a given physical unit.

PRTCT

```
►►──┬──────────────────────┬──────────────────────────────────────►◄
    └─PRTCT=password───────┘
```

statements:	*PU*
dependencies:	*PUTYPE=4 or 5*

Specifies a 1–8 EBCDIC character password used to verify the identity of a caller or receiver when switched subarea connections are being established. If you do not specify this operand, the identity of the caller or receiver is not verified when a connection is established.

The password value must be the same at the SSCP of the caller and the SSCP of the receiver. When one side has the password defined, but the other side does not, or if the caller, receiver, or other SSCP does not support call security verification, the connection fails regardless of which side initiates the call. The NCP must support call security verification.

This operand is valid only when PUTYPE=4 or PUTYPE=5.

PUTYPE

```
►►─┬─PUTYPE=2──────────┬──────────────────────────────────────────►◄
   │                   │
   └─PUTYPE=─┬─2─┬─────┘
             ├─1─┤
             ├─4─┤
             └─5─┘
```

statements:	*GROUP, PU*
dependencies:	*none*

Specifies the physical unit type. The physical unit type depends on the type of cluster controller (such as 3767, 3174, or AS/400). To determine the physical unit type for a given device, see the component description manual for that device.

For a type 2.1 physical unit, code PUTYPE=2.

For switched subarea connections, code PUTYPE=4 or PUTYPE=5. PUTYPE=4 defines the physical unit as an NCP. PUTYPE=5 defines the physical unit as a VTAM.

Note: For PU types 4 and 5, an auto call unit (ACU) is required for subarea dial connections to be established.

The physical unit type for a LAN switched station must be 2.

For a PU type 1 or 2, or type 2.1, the value of PUTYPE occupies bits 4–7 of the 48-bit station ID constructed by VTAM for switched network operation. See “IDNUM” on page 437 for a description of the station ID.

Note: The only valid value for Enterprise Extender is PUTYPE=2.

REDIAL

```
►►─┬─REDIAL=3──────────────────┬──────────────────────────────────►◄
   │                           │
   └─REDIAL=number_of_redials──┘
```

statements:	*PATH*
dependencies:	*none*
range:	*0–254*

Specifies the number of times dialing is to be retried before returning a dialing error to VTAM.

If you code 0, dialing is not to be retried.

REGISTER

For Independent LUs:		For Dependent LUs:	
REGISTER=NO REGISTER= NO / CDSERVR / NETSRVR		REGISTER=NETSRVR REGISTER= NETSRVR / CDSERVR / NO	
statements:	PU, LU	*statements:*	PU, LU
dependency:	APPN only	*dependency:*	APPN only

Specifies how a resource should be registered. The default depends on whether you code REGISTER for dependent or independent LUs.

REGISTER=CDSERVR
Indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same effect as NO.

REGISTER=NETSRVR
Indicates that the end node resource should be registered to its network node server, but that directory registration should not be requested for it.

For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

REGISTER=NO
Indicates that the resource should not be registered.

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

RESETLU

```
►►─┬─RESETLU=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─RESETLU=─┬─YES─┬──────┘
              └─NO──┘
```

statements: *PU, LU*
dependencies: *dependent LUs only*

Specifies whether a DACTLU request followed by a ACTLU request will be sent during clean-up processing.

RESETLU=YES
Specifies that during clean-up processing for dependent LUs, VTAM will issue an ACTLU request after the DACTLU request.

RESETLU=NO
Specifies that during clean-up processing for LU-LU sessions, RESETLU=NO prevents a DACTLU-ACTLU from flowing to an LU.

RESSCB

```
►►─┬─RESSCB=0──────────────────────────────┬──────────────────────────────────►◄
   │                                       │
   └─RESSCB=number_of_session_control_blocks─┘
```

statements:	*LU*
dependencies:	*independent LUs*
range:	*0–65535*

Specifies the number of session control blocks reserved by the NCP for the logical unit. This operand is valid for independent logical units only. If you code it for other logical units, VTAM ignores it.

This operand applies only to type 2.1 peripheral nodes.

For information about NCP restrictions on this operand, refer to the *NCP, SSP, and EP Resource Definition Reference*.

SAPADDR

```
►►─┬─SAPADDR=4───────┬──────────────────────────────────────────────────────►◄
   │                 │
   └─SAPADDR=address─┘
```

statements:	*PATH*
dependencies:	*MEDIUM=HPRIP*

Specifies the service access point for the switched connection through an Enterprise Extender network.

Notes:

1. The SAP values for Enterprise Extender are multiples of 2 and for all other mediums a multiple of 4 applies. A null SAPADDR value has the same meaning as not coding the SAPADDR operand.
2. The remote SAP value specified by SAPADDR must be unique. For more information, see the OS/390 SecureWay Communications Server: SNA Network Implementation Guide
3. Exercise caution when specifying a Switched Major Node which will be active when the connection network function is in effect for HPR/IP. The remote SAP values specified with SAPADDR on the PATH statement must not match local SAP values specified in a node in the HPR/IP Shared Access Transport Facility (SATF) if connections will be attempted through the defined PATH (and PU) while connections through the connection network are active to that node.

SECNET

```
►►─┬─SECNET=NO──────────┬──────────────────────────────────────────►◄
   │                    │
   └─SECNET=─┬─NO──┬────┘
             └─YES─┘
```

statements: *PU*
dependencies: *PUTYPE=1 or 2*

Specifies whether this physical unit is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

This operand applies only if you have IBM 3710 Network Controllers or IBM 3174 token-ring attachments in your network.

SECNET=NO
Data received by VTAM from this physical unit is not flagged as requiring special problem determination considerations when routed to a communication network application.

SECNET=YES
Data that is received by VTAM from this physical unit is flagged as requiring special problem determination considerations when routed to a communication network application.

SECURITY

```
►►─┬─SECURITY=UNSECURE──────────┬──────────────────────────────────►◄
   │                            │
   └─SECURITY=─┬─UNSECURE──┬────┘
               ├─PUBLIC────┤
               ├─UNDERGRO──┤
               ├─SECURE────┤
               ├─GUARDED───┤
               ├─ENCRYPT───┤
               └─SHIELDED──┘
```

statements: *GROUP, PU*
dependencies: *APPN only*

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level.

SECURITY=PUBLIC
Specifies a public switched network.

SECURITY=UNDERGRO
Specifies an underground cable, not guarded.

SECURITY=SECURE
Specifies a secure conduit, not guarded.

SECURITY=GUARDED
Specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
Specifies link encryption.

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded.

SHM

```
►►─┬─SHM=NO──────────┬──────────────────────────────────────────►◄
   │                 │
   └─SHM=─┬─NO──┬────┘
          └─YES─┘
```

statements:	*PATH*
dependencies:	*DIALNO; GRPNM; LINENM; NPSI*

Specifies whether the switched subarea connection made using this PATH definition statement should be a short-hold mode connection.

SHM is mutually exclusive with the LINENM operand. It is valid only for connections established through X.25 NCP Packet Switching Interface (NPSI).

SHM=NO
Specifies that short-hold mode is not to be used.

SHM=YES
Specifies that short-hold mode is to be used, and the value specified by the SHMTIM operand should be used as the short-hold mode timer.

When SHM=YES, the DIALNO and GRPNM operands are required. The GRPNM operand is used to specify which group is appropriate for this PATH definition statement.

SHMTIM

```
►►─┬─SHMTIM=0.0──────┬──────────────────────────────────────────►◄
   │                 │
   └─SHMTIM=time─────┘
```

statements:	*PATH*
dependencies:	*NPSI connections*
range:	*0.0–6553.5 seconds*

Specifies the time, in seconds and tenths of seconds, to be used as the short-hold mode timer for a short-hold mode connection that uses the SHM operand.

If SHMTIM=0.0, a default timer value associated with the line is used. Consult the documentation of the product providing the short-hold mode support for the range of timer values that it supports.

This operand is valid only for connections established through NPSI.

SPAN

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─SPAN=span_name───┘
```

statements: *PU, LU*
dependencies: *NetView*

Defines a span of control for VTAM minor node resources. Code this operand if you are using the NetView program. For a full description, refer to the *TME 10 NetView for OS/390 Installation and Administration Guide*.

The NetView program checks the SPAN value, but VTAM ignores it.

SSCPFM

```
    ┌─SSCPFM=FSS──────────┐
►►──┼─────────────────────┼──────────────────────────────────────────►◄
    │            ┌─FSS────┐
    └─SSCPFM=────┼────────┼──┘
                 ├─USSNTO─┤
                 ├─USSSCS─┤
                 ├─USS3270┤
                 ├─USS3275┤
                 └─USS3780┘
```

statements: *GROUP, PU, LU*
dependencies: *PUTYPE=1 or 2*

Determines what type of RUs the logical unit can support in its communications with the SSCP.

SSCPFM=FSS
Specifies that field-formatted RUs are supported.

SSCPFM=USSNTO
Specifies that this logical unit is supported by NTO in the boundary node, and character-coded RUs are supported over its SSCP-LU sessions.

SSCPFM=USSSCS
Specifies that character-coded RUs are supported.

SSCPFM=USS3270
Allows printing of a user-defined logon message (USSMSG10) on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

SSCPFM=USS3275
Prevents printing of a user-defined logon message (USSMSG10) for a terminal defined as an SDLC 3275 (PU type 1) that might have an attached printer. Code SSCPFM=USS3275 to prevent the user-defined logon message from printing on BSC printers.

SSCPFM=USS3780
Coded for non-SNA devices supported with NTO that use the 3780 protocol.

These devices use USS command facilities such as LOGON and IBMTEST and are supported only by Release 2 of NTO. Refer to the *NTO Installation* for more information.

SUBAREA

```
►►─┬─SUBAREA=inbound_xid_value─┬──────────────────────────────────────►◄
   └─SUBAREA=suba_addr─────────┘
```

statements:	*PU*
dependencies:	*PUTYPE=4 or 5*
range:	*1–value of MXSUBNUM start option*

Specifies the subarea address assigned to the physical unit (NCP or VTAM) represented by this PU definition statement. VTAM compares the subarea number in the partner's XID with the defined subarea number to verify that the node on the other end of the switched subarea connection is the one that was expected. This operand is required when PUTYPE=4 or PUTYPE=5.

The SUBAREA value must be the same as the subarea address defined for this physical unit in the other domains of the network. This operand is for switched subarea connections.

TERM

```
►►─┬──────────────┬──────────────────────────────────────────────────►◄
   └─TERM=type────┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*

Identifies, for a virtual logical unit supported through NTO, the device data stream compatible characteristics. An application program can find these characteristics by executing an INQUIRE DEVCHAR macroinstruction that names this virtual logical unit. The information is placed in the DEVAUXTP field of the DEVCHAR DSECT. (DEVAUXTP is an 8-byte storage area provided by the application program.) For more information on the DEVCHAR DSECT, refer to *OS/390 eNetwork Communications Server: SNA Programming*

TGCONF

```
►►─┬─TGCONF=(MULTI,NORMAL)──────────────────────────────────────────────┬──►◄
   │                                                                    │
   └─TGCONF=(──┬─────────┬──,──┬──────────────────────────────────┬──)──┘
               ├─MULTI───┤     ├─NORMAL─────────────────────────┤
               └─SINGLE──┘     └─STANDBY──,──┬──────────┬──────┘
                                             ├─nn───────┤
                                             └─NONE─────┘
```

statements:	*PU*
dependencies:	*PUTYPE=4*

Specifies whether this subarea link station is in a multilink or single-link transmission group. TGCONF is valid only if you specify PUTYPE=4.

Multilink Transmission Groups: Code TGCONF=MULTI or omit TGCONF for all subarea link stations in a multilink transmission group.

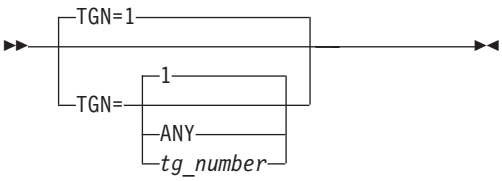
Single-Link Transmission Groups: You do not need to code TGCONF for a single-link transmission group. However, coding TGCONF=SINGLE for a transmission group that contains a single line can improve NCP performance for the transmission group.

Standby Stations: TGCONF=STANDBY designates if a subarea station is to be used in hot-standby mode. TGCONF=STANDBY is not valid when TGCONF=SINGLE is coded.

Congestion Relief: TGCONF=(MULTI,STANDBY,nn) designates that a subarea station is to be used for hot-standby and congestion relief. The subarea transmits data to relieve congestion in the transmission group configuration if queued data reaches the value specified by nn. nn indicates the percentage reached of a threshold specified by the totthresh suboperand of the ERn keyword on the associated PATH definition statement. The totthresh suboperand defines the total number of bytes that can be queued for transmission to an MLTG before NCP takes control measures to slow down the TG traffic. Valid values for nn are 3, 6, 12, 25, 50, 75, 87, 93, 97, and 100. NONE designates that the station is used only for hot-standby and not for congestion relief.

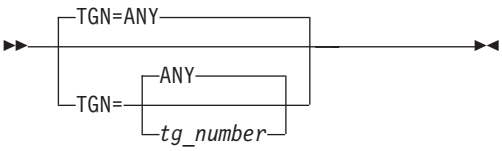
TGN

For subarea TGN:



statements:	PU
dependency:	PUTYPE=4,5
range:	1–255

For APPN TGN:



statements:	PU
dependency:	PUTYPE=2, CPNAME, NETID
range:	0–20

Subarea: Specifies the transmission group (TG) of the switched subarea connection associated with this physical unit. For subarea, this operand is valid only when PUTYPE=4 or PUTYPE=5.

APPN: Specifies the transmission group (TG) that represents the connection between this node and the destination APPN node. For APPN, this operand is valid when PUTYPE=2.

By specifying a *tg_number*, you indicate a preferred TG which might or might not be used depending on how the partner nodes negotiate the number.

For APPN-capable nodes, the CPNAME and NETID operands are required with the TGN operand. Note that you can specify the CPNAME operand without the TGN operand and the transmission group number will be negotiated. However, if a value for TGN is specified, then a value for CPNAME and NETID must also be specified.

TGN=*tg_number*
Specifies a decimal number assigned to the transmission group.

For APPN, coding TGN=0 means that the PU does not support parallel TGs.

TGN=ANY
For subarea, allows the adjacent node to accept whatever valid transmission group number it receives on the XID. You can specify TGN=ANY for only one end of the transmission group. For the other end, specify a valid TGN number or let it default to 1. If you specify TGN=ANY for both ends, an error occurs. For APPN, coding TGN=ANY, or not coding TGN, allows the transmission group number to be negotiated. For connections to LEN nodes, or to any node that does not support parallel TGs, TGN=0 is always assumed.

Note: VTAM does not validate TGN when coded for subarea resources.

TGP

```
►►──┬──────────────┬──────────────────────────────────────────►◄
    └TGP=tgp_name──┘
```

statements: *GROUP, PU*
dependencies: *APPN only*

Specifies the name of a transmission group (TG) profile. The characteristics of the TG profile (along with any modifiers) become the characteristics of the PU. If TGP is not specified or has not been activated when the PU becomes active, default TG characteristics are assigned.

See “APPN Transmission Group Profile” on page 513 for more information on transmission group profiles.

TOPO

```
      ┌─TOPO=CONNECT──────────────┐
►►────┼───────────────────────────┼──────────────────────────►◄
      │           ┌─CONNECT───────┐│
      └─TOPO=─────┼───────────────┼┘
                  │─connection_cost─│
                  └─NONE──────────┘
```

statements: *PU*
dependencies: *APPN only*
range: *0–255*

Controls the way APPN connections using this link station will be reported to APPN topology and routing services for inclusion into APPN functions.

If you change a VTAM’s node type from network node to end node by inactivating the VTAM and then reactivating it with NODETYPE=EN, you might need to change the switched PU definitions in partner VTAMs to use TOPO=NONE. This is suggested only if the switched PU connection is predefined and you have been using this VTAM as an intermediate routing node.

TOPO=CONNECT
APPN topology and routing services receives information about the connection and its characteristics when the connection is activated. When the connection is deactivated, APPN topology and routing services receives notification that the TG is inactive.

TOPO=*connection_cost*
Indicates that the connection is to be treated as auto-active with a connection cost override. Auto-active allows a switched physical unit to be selected for routing prior to the actual establishment of the connection.

If a *connection_cost* is coded, then CPNAME, NN, and TGN=*tg_number* must also be coded. If one or more of these operands is not coded, then the connection cannot be reported prior to activation and the TOPO support is forced to CONNECT.

Note: *connection_cost* is valid only for switched major nodes.

TOPO=NONE
APPN topology and routing services receives no information concerning connections established with this link station. This means that connections using this link station are not available for APPN network services.

TYPE

```
►►──TYPE=SWNET──────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Means that the VBUILD definition statement defines a switched major node to VTAM. All physical units defined in this major node can be connected only by a switched link. This operand is required.

UPARM1

```
      ┌─UPARM1=128──────────────────────┐
►►────┤                                 ├──────────────────────────────►◄
      └─UPARM1=user-defined_operand_value─┘
```

statements:	*GROUP, PU*
dependencies:	*APPN only*
range:	*0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
►►─┬─UPARM2=128──────────────────────────┬──────────────────────────────────────►◄
   │                                     │
   └─UPARM2=user-defined_operand_value───┘
```

statements: *GROUP, PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────────────►◄
   │                                     │
   └─UPARM3=user-defined_operand_value───┘
```

statements: *GROUP, PU*
dependencies: *APPN only*
range: *0–255*

Specifies a user-defined operand value.

USE

```
►►─┬─USE=YES──────────┬──────────────────────────────────────────────────────►◄
   │                  │
   │          ┌─YES──┐│
   └─USE=─────┤      ├┘
              └─NO───┘
```

statements: *PATH*
dependencies: *none*

Specifies whether the path is initially usable or not usable. This attribute of the path can be modified by the VTAM operator, using the VARY PATH command. The effect of USE=YES and USE=NO for a path is similar to the effect of ISTATUS=ACTIVE and ISTATUS=INACTIVE for a minor node.

USE=NO
Specifies that the path is initially not usable.

USE=YES
Specifies that the path is initially usable.

USSTAB

```
►►─┬─USSTAB=ISTINCDT──────────┬──────────────────────────────────────────────►◄
   │                          │
   └─USSTAB=uss_table_name────┘
```

statements: *GROUP, PU, LU*
dependencies: *PUTYPE=1 or 2; dependent LUs only*

Specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

A terminal user can issue a USS command with the LANGTAB operand. This causes a second USS table to be associated with the logical unit, which overrides the table specified with USSTAB. If you do not code USSTAB and a LANGTAB USS table is not in use, the IBM-supplied USS table (ISTINCDT) is used. For more information on USS tables, see “Unformatted System Services (USS) Tables” on page 731.

USSTAB is valid for dependent logical units only.

VERALSID

```
                 ┌─VERALSID=NO──────────┐
►►───┬──────────┼──────────────────────┼──────────────────────────────────►◄
     │          │                      │
     │          └─VERALSID=─┬─NO──┬────┘
     │                      └─YES─┘
     │          ┌─VERALSID=NO──┐   (1)
     └──────────┼──────────────┼──────────────────────────────────────────
                └─VERALSID=NO──┘
```

Notes:

1 VERALSID=NO is the only valid value if the TRLE operand is coded.

statements: *GROUP, PU*
dependencies: *APPN only*

Specifies whether the adjacent link station name should be used to determine the link station. When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to the XID3. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

VERALSID=NO
Specifies that the adjacent link station name is not used to determine the link station.

VERALSID=YES
Specifies that the adjacent link station name is used to determine the link station.

When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to XID3s. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under a nonswitched line.

VERID

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    └─VERID=──┬─call_identifier──────────┬─┘
              └─(─call_identifier──────)─┘
```

statements: *PATH*
dependencies: *GRPNM*

Specifies one or more call identifiers to be used by call ID verification.

Call identifier is a general term for an identifier provided by a switched network that identifies the telephone number of the device at the other end of the switched line. For example, the calling DTE address is such an identifier in X.25 networks. Each call identifier is specified as a character string and can be up to 251 bytes long.

If you code VERID:
- GRPNM is required.
- VERIFY=NONE is *not* valid.
- The *list form* is required if you specify more than one call identifier. For example, to specify call identifiers *id1* and *id2*, code VERID=(id1,id2).

VERIFY

```
    ┌─VERIFY=NONE──────────┐
►►──┼──────────────────────┼──────────────────────────────────────────────────►◄
    │            ┌─NONE──┐ │
    └─VERIFY=────┼───────┼─┘
                 ├─IN────┤
                 ├─INOUT─┤
                 └─OUT───┘
```

statements: *PATH*
dependencies: *GRPNM; VERID*

Specifies when call ID verification is to be performed if this PATH definition statement is used.

If the VERIFY operand is specified or defaults to NONE and all PATH definition statements for this physical unit specify or default to CALL=INOUT, an inbound call is accepted from a switched line group.

If call ID verification (VERIFY=IN, VERIFY=OUT, or VERIFY=INOUT) is specified on any PATH definition statement, an inbound call is accepted only if it is using a line group for which there is a PATH definition statement that specifies or defaults to CALL=IN or CALL=INOUT. That is, if one PATH definition statement specifies call ID verification, all valid paths (inbound and outbound) must be defined, including those paths for which call ID verification is not to be performed.

If you code the VERIFY operand with any value other than VERIFY=NONE, GRPNM is required.

VERIFY=IN
Specifies that call ID verification is to be performed for inbound calls using this

PATH definition statement. If VERIFY=IN, CALL=OUT is not valid. If VERIFY=IN, VERID and GRPNM are required.

VERIFY=INOUT
Specifies that call ID verification is to be performed for inbound and outbound calls using this PATH definition statement. If VERIFY=INOUT, CALL=INOUT is required. If VERIFY=INOUT, VERID and GRPNM are required.

VERIFY=NONE
Specifies that call ID verification is not to be performed for inbound or outbound calls using this PATH definition statement. If VERIFY=NONE, VERID is not valid.

VERIFY=OUT
Specifies that call ID verification is to be performed for outbound calls using this PATH definition statement. If VERIFY=OUT, CALL=IN is not valid. If VERIFY=OUT, VERID and GRPNM are required.

VPACING

```
►►─┬─VPACING=2──────────────────────┬──────────────────────────────────────────►◄
   │                                │
   └─VPACING=─┬─2──────────────────┬┘
              └─number_of_requests─┘
```

statements:	*GROUP, PU, LU*
dependencies:	*PUTYPE=1 or 2*
range:	*0–63*

Specifies how VTAM paces the flow of data from the boundary node containing the PLU to the boundary node that performs pacing for a channel-attached SNA device.

Note: You can also use the FIXED option on this operand. FIXED is an NCP option that allows you to override adaptive session pacing and use fixed pacing instead.

To use the FIXED option, you must code (N,,FIXED). The second comma represents a minimum value that is used by some pre-ENA nodes.

For more information on pacing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

VTAMTOPO

```
►►─┬─VTAMTOPO=REPORT────────┬──────────────────►◄
   │                        │
   └─VTAMTOPO=─┬─REPORT───┬─┘
               ├─NOREPORT─┤
               ├─INCLUDE──┤
               └─IGNORE───┘
```

statements: VBUILD
dependency: none

```
►►──┬──────────────────────────────────┬──────────►◄
    │                ┌─REPORT───┐      │
    └─VTAMTOPO=──────┤          ├──────┘
                     └─NOREPORT─┘
```

statements: GROUP or PU
dependencies: none

Specifies whether the VTAM topology agent reports the major nodes or subordinate minor nodes to the CMIP manager application programs for SNA local topology.

Specifying VTM.VTAMTOPO allows VTAMTOPO to be handled appropriately by the NCP Definition Facility for versions of NDF that do not recognize VTAMTOPO natively.

Note: Refer to “Appendix C. VTAMTOPO Filtering Option Reporting” on page 837 for more information on VTAMTOPO filtering option reporting.

VTAMTOPO=REPORT

On VBUILD: Specifies that the VTAM topology agent reports subordinate PUs. This does not override any value specified on the GROUP or PU.

On GROUP: Specifies that the VTAM topology agent reports subordinate PUs. This does not override any value specified on the PU.

On PU: Specifies that the VTAM topology agent reports this PU.

VTAMTOPO=NOREPORT

On VBUILD: Specifies that the VTAM topology agent does not report subordinate PUs. This major node is not reported by itself. This does not override any value specified on the GROUP or PU.

On GROUP: Specifies that the VTAM topology agent does not report any subordinate PUs. This does not override any value specified on the PU.

On PU: Specifies that the VTAM topology agent does not report this PU.

VTAMTOPO=INCLUDE

On VBUILD: Specifies that the VTAM topology agent reports all subordinate PUs. VTAMTOPO=INCLUDE overrides any other value specified on the GROUP or PU.

VTAMTOPO=IGNORE

On VBUILD: Specifies that the VTAM topology agent does not report any subordinate PUs. VTAMTOPO=IGNORE overrides any other value specified on the GROUP or PU.

XNETALS

```
     ┌─XNETALS=value_of_XNETALS_start_option─┐
►►───┤                                       ├──────────────────────────────────►◄
     └─XNETALS=──┬─NO──┬──────────────────────┘
                 └─YES─┘
```

statements: *PU*
dependencies: *none*

determines whether the network ID of an attaching node or VTAM's network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true. If the NETID that the PU sent in its XID is not the same as VTAM's and XNETALS=NO or is defaulted, IST680I will be issued.

XNETALS=NO
Indicates that VTAM's network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement, then it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement, then VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES
indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Note: Providing that XNETALS=YES is coded, you can prohibit nonnative network attachment for LU-LU sessions for a particular PU by coding XNETALS=NO on the GROUP, LINE or PU statements.

Transport Resource List Major Node

To define a transport resource list major node, code the following definition statements:

- One VBUILD definition statement to begin the transport resource list major node.
- One transport resource list element (TRLE) definition statement for each multipath channel (MPC) connection.

The transport resource list element is not a resource, but describes the connectivity characteristics of the multipath channel line that is used for the connection.

An MPC connection can be used for both SNA & TCP/IP Open Edition application traffic. Refer to *OS/390 SecureWay Communications Server: IP Configuration*. Any user of the connection is referred to as an upper layer protocol (ULP).

For a connection between VTAM and an adjacent APPN node, the TRLE operand on the PU definition statement in the local SNA major node that defines the adjacent APPN node identifies the TRLE definition statement VTAM uses to route data over the connection.

For a connection between VTAM and a port on an IBM S/390 Open Systems Adapter through which native access to an ATM network is achieved, the PORT definition statement in the XCA major node that defines the port identifies the TRLE definition statement VTAM uses to route data over the connection. The determination of which TRLE definition statement is used is made by matching the port name specified on the PORTNAME operand on the PORT definition statement to the port name specified on the PORTNAME operand on a TRLE definition statement in the TRL major node.

For all MPC connections defined via a TRLE definition, the subchannel addresses for the MPC connection must be defined. For all connections except OSA-Express, each subchannel address is dedicated to one direction of flow, either read or write, and the addresses are defined on the READ and WRITE operands.

For OSA-Express, there are two types of subchannels. There are subchannels dedicated to control flows, and there are subchannels dedicated to data. Control flows can be either the setting up and terminating of connections or any data the ULP identifies as control. Data is any data the ULP identifies as not control. The control subchannels are defined on the READ and WRITE operands, and the data subchannels are specified on the DATAPATH operand. As with IBM S/390 Open Systems Adapter ATM access, the PORTNAME operand on the TRLE definition statement is used by TCP/IP for TCP/IP to achieve connectivity through the OSA-Express.

Transport Resource List Major Node

Note: For OSA-Express, Channel Unit Address and Subchannel are the same as the device number as specified in the hardware configuration definitions.

For detailed instructions on setting up an OSA-Express feature, refer to the *S/390 Open Systems Adapter-Express Customer's Guide and Reference*.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about MPC connections.

Table 34. Transport Resource List Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=TRL	478	R			
name	TRLE	DATAPATH	472				
		LNCTL	472	R		MPC	
		MAXBFRU	472			5	
		MAXREADS	473			4	MPCLEVEL=HPDT
		MPCLEVEL	474			HPDT	HPR start option, MAXREADS, STORAGE
		MPCUSAGE	474			SHR	MPCLEVEL=HPDT
		PORTNAME	475				ATM native connections
		READ	476	R			
		REPLYTO	477			3.0	
		STORAGE	477			DS	MPCLEVEL=HPDT
		WRITE	478	R			

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement for each TRL major node.

TRLE Definition Statement

Code one TRLE definition statement for each MPC line that is used for a connection.

Full Syntax

The full syntax for the transport resource list major node follows. The syntax for each operand is repeated in the "Operand Description" section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=TRL──────────────────────────────────────────►◄
    └name──┘
```

TRLE

```
                                   ┌──────────────────────────────┐
                                   ▼                              │
►►──name──TRLE──┬──────────────────┬──┬──────────────────────────┬──►
                └──DATAPATH──=─────┘  └──(channel_unit_address)──┘

                          (2)
              ┌──MAXBFRU=5──────────────────┐
                          (1)
              ├──MAXBFRU=1──────────────────┤
►──LNCTL=MPC──┤                             ├──────────────────────────►
              └──MAXBFRU=number_of_buffers──┘

  ┌──MAXREADS=4──────────────────────────┐   ┌──MPCLEVEL=HPDT──────────┐
►─┤                                      ├───┤                         ├──►
  └──MAXREADS=number_of_available buffers┘   └──MCPLEVEL=──┬──HPDT───┬──┘
                                                           ├──NOHPDT─┤
                                                           └──QDIO───┘

  ┌──MPCUSAGE=SHR──────┐
►─┤                    ├──PORTNAME=(port_name,link_num)──────────────►
  └──MCPUSAGE=──┬──EXC─┬──┘
                └──SHR─┘

                                   ┌──REPLYTO=3.0──────────┐
►──READ=(──channel_unit_address──)─┤                       ├──────────►
                                   └──REPLYTO=timeout_value┘

  ┌──STORAGE=DS──────────┐
►─┤                      ├──WRITE=(──channel_unit_address──)──────►◄
  └──STORAGE=──┬──DS────┬──┘
               └──ESCA──┘
```

Notes:

1 MAXBFRU defaults to 1 when HPR=NO, XID=NO, CONNTYPE=LEN, or the VTAM node is not HPR capable.

2 MAXBFRU defaults to 5 when HPR=YES, XID=YES, CONNTYPE=APPN, and the VTAM node is HPR capable.

Operand Descriptions

The following section describes the operands you can code for the transport resource list major node.

DATAPATH

```
                  ┌─,──────────────────────────┐
                  ▼                            │
►►─┬─DATAPATH──=─┴─┬─(channel_unit_address)─┴─┬──────────────────────────►◄
```

statements:	*TRLE*
range	*X'000'-X'FFFF'*

Specifies the subchannel addresses used to read and write data through an OSA-Express connection. You must code at least one DATAPATH subchannel. Each TCP/IP instance that issues a START DEVICE for an OSA-Express feature will be assigned one of the DATAPATH channels by VTAM. Sufficient DATAPATH subchannel addresses must be coded for the number of concurrent instances that will be using an OSA-Express port.

Subchannel addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen(-). VTAM generates addresses for the range. For example, DATAPATH=(132,128-130,13A) defines the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A. There is a limit of 238 addresses for this operand.

Do not code the same subchannel addresses for the DATAPATH operand as was coded for either the READ operand or the WRITE operand in the same host.

LNCTL

```
►►─LNCTL=MPC──────────────────────────────────────────────────────────────►◄
```

statements:	*TRLE*

Indicates that the link is a multipath channel link.

This operand is required.

MAXBFRU

```
                 (2)
     ┌─MAXBFRU=5──────────────────┐
                 (1)             │
     ├─MAXBFRU=1──────────────────┤
►►───┤                            ├──────────────────────────────────────►◄
     └─MAXBFRU=number_of_buffers──┘
```

Notes:

1 MAXBFRU defaults to 1 when HPR=NO, XID=NO, CONNTYPE=LEN, or the VTAM node is not HPR capable.

2 MAXBFRU defaults to 5 when HPR=YES, XID=YES, CONNTYPE=APPN, and the VTAM node is HPR capable.

statements:	*TRLE*
range:	*1–16*

Specifies the number of 4K buffer pages VTAM uses to receive data when activating a multipath channel. The same MAXBFRU value is used for each of the READ subchannels. The resulting buffer size is *number_of_pages* multiplied by 4K. The total buffer space used is 4K multiplied by *number_of_pages* multiplied by the number of READ subchannels.

VTAM automatically substitutes a value of 16 for any coded value higher than 16 without issuing a warning message.

Some devices might have a hardware restriction on the maximum number of buffers. If there is a hardware buffer size for multipath channel, VTAM compares the MAXBFRU value and the hardware buffer size and uses the smaller of the two to avoid excessive fixed storage.

MAXBFRU is not used to determine the buffer size of the WRITE subchannels coded on this host. The WRITE subchannel buffer size is determined dynamically by VTAM, depending on the setting at the READ end of the channel.

You can reduce this value after examining VTAM multipath channel tuning statistics so that excessive storage is not allocated for each subchannel. To obtain tuning statistics output, code the TNSTAT start option.

For connections to an OSA-Express feature, MAXBFRU is fixed at 2 and should not be coded.

MAXREADS

```
►►─┬─MAXREADS=4──────────────────────────────┬──────────────────────────────────►◄
   │                                         │
   └─MAXREADS=number_of_available buffers────┘
```

statements:	*TRLE*
range:	*4–8*
dependencies:	*MPCLEVEL=HPDT*

Specifies how many read buffers are available to read data over a high performance data transfer (HPDT) MPC connection.

This operand is valid only when MPCLEVEL=HPDT is specified.

Note: The number of read buffers on the TRLE definition statement defining a port on an IBM S/390 Open Systems Adapter through which native access to an ATM network is achieved should always be defined to eight (MAXREADS=8.)

For connections to an OSA-Express feature, MAXREADS is fixed at 2 and should not be coded.

MPCLEVEL

```
   ┌─MPCLEVEL=HPDT──────────┐
►►─┤                        ├──────────────────────────────────────────►◄
   └─MCPLEVEL=──┬─HPDT───┬──┘
                ├─NOHPDT─┤
                └─QDIO───┘
```

statements: *TRLE*
dependencies: *HPR start option, MAXREADS, STORAGE*

Specifies the level of MPC capability to be used for the connection. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about the different levels of MPC capability provided by VTAM.

MPCLEVEL=HPDT
Indicates that HPDT MPC is to be used for the connection. This operand is required for native ATM connections.

For connections to an adjacent APPN node or an IBM S/390 Open Systems Adapter port: When MPCLEVEL=HPDT, you must code HPR=RTP in the start list for this VTAM. This restriction does not apply if this MPC connection is only for IP traffic.

MPCLEVEL=NOHPDT
Indicates that non-HPDT MPC is to be used for the connection. If MPCLEVEL=NOHPDT is specified and a value is coded for the MAXREADS or STORAGE operand, the values specified for MAXREADS and STORAGE are ignored.

MPCLEVEL=QDIO
Indicates that the Queued Direct I/O interface is used to an OSA-Express feature. Control data is transmitted across the one READ and one WRITE device. Normal data is transmitted across a DATAPATH device, where each ULP is assigned its own device number. When MPCLEVEL is QDIO, HPDT MPC is also used for this connection across the control channels.

MPCUSAGE

```
   ┌─MPCUSAGE=SHR───────┐
►►─┤                    ├──────────────────────────────────────────►◄
   └─MCPUSAGE=──┬─EXC─┬──┘
                └─SHR─┘
```

statements: *TRLE*
dependencies: *MPCLEVEL=HPDT*

Specifies whether multiple usage (sharing) or single usage (exclusive) of an MPC (with HPDT) group is allowed. This keyword is ignored for non-HPDT capable MPC groups. Users of the MPC group are the various protocol stacks (for example, APPN or TCP/IP).

MPCUSAGE=SHR
Indicates this group can be used by multiple protocol stacks at the same time.

MPCUSAGE=EXC
Indicates this group can only be used by a single protocol stack at a time. The first to activate the group becomes the exclusive user of that MPC group. All subsequent activation attempts (while the group is in use) will be rejected. Defining exclusive usage will provide improved performance.

name

```
►►──┬──────┬──────────────────────────────►◄   ►►──name──────────────────────────────►◄
    └name──┘
```

statements:	VBUILD	*statements:*	TRLE

On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

On TRLE: Assigns a required name that identifies the TRLE definition statement. This name cannot match the VTAMLST member name containing this Transport Resource List Major Node.

For connections to an IBM S/390 Open Systems Adapter for native ATM access:

Name you specify on the TRLE definition statement must match the name of the IBM S/390 Open Systems Adapter. The name of the IBM S/390 Open Systems Adapter is defined during IBM S/390 Open Systems Adapter configuration on the ATM Native Support Settings panel, which is part of the OSA Configuration pull down option accessed through the OSA/SF interface.

For connections to an OSA-Express for GIGABIT Ethernet:

Name you specify on the TRLE definition statement defines the name of the IBM S/390 OSA-Express for use by this VTAM.

PORTNAME

```
►►──PORTNAME=(port_name,link_num)──────────────────────────────────────────────►◄
```

statements:	*TRLE*
dependencies:	*MPCLEVEL=HPDT or QDIO, Native ATM connections and OSA-Express connections*

Specifies the name of a port on an IBM S/390 Open Systems Adapter through which native access to an ATM network is achieved, or the name of an

OSA-Express through which GIGABIT ethernet access is achieved. If defined for native ATM access, the name of the physical port is the name as it is defined during IBM S/390 Open Systems Adapter configuration, which is part of the OSA Configuration pull down option accessed through the OSA/SF interface. No link_num should be defined for this configuration. If defined for an OSA-Express feature, the name of the physical port is configured only by this Transport Resource List Entry. There is no required OSA/SF matching configuration. The link_num maps a logical link number to this PORTNAME, and is likewise configured only by this Transport Resource List Entry. If not defined, the link_num defaults to 0.

The maximum length for PORTNAME is 8 characters.

Note: If there is more than one TRLE used for the same OSA, specify different NAME parameters in the TRLE statements.

READ

```
►►──READ=(─channel_unit_address──)──────────────────────────────────────►◄
```

statements:	*TRLE*
range:	*X'000'–X'FFFF'*

Specifies the subchannel address used to read data from the adjacent APPN node or IBM S/390 Open Systems Adapter port. You must code at least one READ subchannel.

For an OSA-Express feature, it specifies the subchannel address used to read MPC or ULP specific control data. You must code exactly one READ subchannel.

Subchannel addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen(-). VTAM generates addresses for the range. For example, READ=(132,128-130,13A) defines the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A.

For connections to an adjacent APPN host node:

each address on the READ operand, code the corresponding address on the WRITE operand in the adjacent host to provide a complete path.

For connections to an IBM S/390 Open Systems Adapter port or an OSA-Express port:

you can specify only one subchannel address and that address must be an even number that is one less than the address specified on the corresponding WRITE operand. For example, if you specify 500 on the READ operand, you must specify 501 on the WRITE operand.

If this IBM S/390 Open Systems Adapter port is for native ATM access, the last two digits of the subchannel address you specify on the READ operand must match the Even Unit Address as it is defined during IBM S/390 Open Systems Adapter configuration on the ATM Native Support Settings panel, which is part of the OSA Configuration pull down option accessed through the OSA/SF interface.

If this IBM S/390 Open Systems Adapter port is for an OSA-Express feature, there is no matching configuration required.

For an adjacent APPN node or an IBM S/390 Open Systems Adapter port, do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

REPLYTO

```
►►─┬─REPLYTO=3.0──────────────┬──────────────────────────────────────►◄
   └─REPLYTO=timeout_value────┘
```

statements: *TRLE*
range: *0.1–25.5*

Specifies how long VTAM waits for completion of a multipath channel XID I/O operation after starting a channel program. If this timeout expires, a message is written to inform the operator that a timeout has occurred. After the XID completes, REPLYTO has no meaning.

Specify *timeout_value* in seconds or seconds and tenths of a second. A value of 3.0 seconds should be adequate, unless the attached adjacent APPN node or IBM S/390 Open Systems Adapter port is disabled for long intervals.

For connections to an OSA-Express feature, the REPLYTO parameter only applies to the control channels.

STORAGE

```
►►─┬─STORAGE=DS───────────────┬──────────────────────────────────────►◄
   │                          │
   └─STORAGE=─┬─DS───┬─────────┘
              └─ESCA─┘
```

statements: *TRLE*
dependencies: *MPCLEVEL=HPDT*

Specifies whether CSM ECSA or CSM data space is used for the multipath channel's inbound I/O storage.

This operand is valid only when MPCLEVEL=HPDT is specified.

For connections to an OSA-Express feature, the STORAGE parameter does not apply.

HPDT Users: application attempts to receive the data using OPTCD=XBUFLST (HPDT receive), VTAM moves the data once if the application's preference for incoming data does not match this STORAGE value. The LU 6.2 application can indicate its preference for ECSA, data space, or both by using the XBUFLST-Receive vector (ISTAPC82), which is located in the VTAM-APPCCMD vector list (ISTAPCVL). See *OS/390 eNetwork Communications Server: SNA Programming* for more information.

STORAGE=DS
Indicates that the data segment of the inbound buffer area is to reside in CSM data space pages.

STORAGE=ECSA
Indicates that the data segment of the inbound buffer area is to reside in CSM ECSA pages. This option is only recommended if you have a large number of HPDT applications requesting ECSA storage for incoming data traffic.

TYPE

```
►►──TYPE=TRL──────────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*

Specifies that the VBUILD statement defines a TRL major node to VTAM.

WRITE

```
►►──WRITE=(─channel_unit_address──)──────────────────────────────────►◄
```

statements:	*TRLE*
range:	*X'000'–X'FFFF'*

Specifies the subchannel address used to write data to the adjacent APPN node or IBM S/390 Open Systems Adapter port. Code at least one WRITE subchannel address.

For an OSA-Express feature, it specifies the subchannel address used to write MPC or ULP specific control data. Code exactly one WRITE subchannel address.

Subchannel addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen(-). VTAM generates addresses for the range. For example, WRITE=(132,128-130,13A) defines the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A.

For connections to an adjacent APPN host node:

For each address on the WRITE operand, code the corresponding address on the READ operand in the adjacent host to provide a complete path.

The WRITE subchannels are used in order, so certain subchannels can be given a higher priority by coding them first. For example, you might want to code subchannels with a higher bandwidth first.

For connections to an IBM S/390 Open Systems Adapter or an OSA-Express feature:

You can specify only one subchannel address and that address must be an odd number that is one greater than the address specified on the corresponding READ operand. For example, if you specify 500 on the READ operand, you must specify 501 on the WRITE operand.

For an adjacent APPN node or an IBM S/390 Open Systems Adapter port, do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

For an adjacent APPN node or an IBM S/390 Open Systems Adapter port, the WRITE subchannels are used in order, so certain subchannels can be given a higher priority by coding them first. For example, you might want to code subchannels with a higher bandwidth first.

Chapter 3. Routing and Dynamic Reconfiguration

Routing and Dynamic Reconfiguration

About This Chapter

This chapter describes the definition statements that are used to define the following functions:

- VTAM routes
- Dynamic path update
- Dynamic reconfiguration and dynamic change
- Adjacent SSCP tables
- Adjacent cluster routing definitions
- APPN transmission group profiles
- APPN-to-subarea COS mapping
- Border node COS mapping
- Network node server lists.

VTAM Routes

PATH definition statements are representations of the routes VTAM takes to communicate with other subarea nodes. One or more PATH definition statements are filed in the VTAM definition library. The name assigned to a path definition set is used to activate that set of paths. More than one path definition set can be filed and activated. Note that the VBUILD definition statement is not used.

Communication with other subarea nodes is not possible unless all the required path tables are active. Therefore, you should activate any required path tables before you activate any NCP or channel-attachment major nodes.

Note that, unlike NCP, VTAM requires you to code a reverse virtual route in the VTAM that receives an activate virtual route command because VTAM cannot dynamically allocate a virtual route when it receives the command.

Coding Notes

PATH Definition Statement: Code a PATH definition statement to define routes.

Table 35. Definition Statement for Routes

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PATH	DESTSA	482	R			MXSUBNUM
		DELETER	482				see description
		ER0–ER15	483				
		IPRESOLV	439				
		VR0–VR7	483				
		VRPWS00–72	484				

Legend:

R Required operand.

© Copyright IBM Corp. 1984, 1999

Full Syntax

The full syntax for VTAM routes follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

PATH:

```
►►──┬──────────┬──PATH──DESTSA=──┬─subarea_number──────────────┬──────────────────────────────►
    └─name─────┘                 └─(─subarea_number──────────)─┘

►──┬──────────────────────────────────────────────────────────┬──┬──────────────────────────┬──►
   └─DELETER=──┬─explicit_route_number──────────┬──┘   └─ERn=──┬─(adjsub)──────────┬──┘
               └─(─explicit_route_number──────)─┘              └─(adjsub,tgn)──────┘

►──┬──────────────────┬──┬──────────────────────────────────────────────┬──────────────────►◄
   └─VRn=er_number────┘  └─VRPWSnn=─(minimum_size,maximum_size)──────────┘
```

Operand Descriptions

The following section describes the operands you can code for VTAM routes.

DELETER:

```
►►──┬──────────────────────────────────────────────────────────┬──────────────────────────►◄
    └─DELETER=──┬─explicit_route_number──────────┬──┘
                └─(─explicit_route_number──────)─┘
```

statement:	*PATH*
dependencies:	*see note*

explicit_route_number is the explicit route number for the route to be deleted. The value of *explicit_route_number* can be ER0–ER15. Up to 16 explicit route numbers can be specified in DELETER.

VTAM deletes the specified explicit route from its routing table if the explicit route is not currently operative. If the requested explicit route is currently operative, the request is rejected and a warning message is displayed.

Note: Use extreme caution when coding this operand. Deleting one explicit route might make several routes unusable because multiple origin subareas can use the same routing table entry. Use the DELETER operand only when there is no other method of correcting errors or resolving problems. If the explicit route must be redefined, you should use the add or replace function.

DESTSA:

```
►►──DESTSA=──┬─subarea_number──────────────┬──────────────────────────────────────────►◄
             └─(─subarea_number──────────)─┘
```

statement:	*PATH*

dependencies: *MXSUBNUM*

Specifies the destination subarea number for the routes being defined to subarea nodes in your network. You can code more than one destination subarea as long as the route definition statement holds for each one.

The address of the destination subarea must not exceed the subarea addressing supported by the SSCP (as defined by the MXSUBNUM start option).

ER0-ER15:

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └ERn=──┬──(adjsub)──────────┘
           └──(adjsub,tgn)──┘
```

statement:	PATH
dependencies:	none
adjsub range:	1–value of MXSUBNUM
tgn range:	1–255

Names the adjacent subarea and, optionally, the transmission group (TG) number for the associated explicit routes originating in the host and leading to the destination subarea.

At least one ER must be specified on the PATH definition statement. You can change the ER value when the ER is inactivated or undefined.

adjsub
The subarea number (in decimal) of the adjacent subarea to which traffic should be sent from the host subarea for the explicit route being defined.

tgn
An optional transmission group number for the explicit route being defined.

If you do not specify a value, the default is 1.

name:

```
►►──┬──────────┬──────────────────────────────────────────────────────────►◄
    └name──────┘
```

statement:	*PATH*
dependencies:	*none*

Provides the optional name of the PATH represented by this definition statement.

name is not checked by VTAM for validity, but its use is strongly recommended. Its purpose is to point out the particular PATH definition statement if a definition error or warning message occurs during activation.

VR0-VR7:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─VRn=er_number────┘
```

statement: *PATH*
dependencies: *none*

Associates the virtual route with the explicit route.

er_number is the explicit route number to which the virtual route is mapped. You can change VR to ER mapping after initial activation of a path definition when the VR is inactivated or undefined.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72:

```
►►──┬──────────────────────────────────┬──────────────────────────►◄
    └─VRPWSnn=─(minimum_size,maximum_size)──┘
```

statement: *PATH*
dependencies: *none*

Specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition. You can change the pacing window size after initial activation of a path definition when the VR is inactivated or undefined.

Virtual route pacing window sizes are taken from the PATH definition statements in the node that activates the virtual route. For instance, you could define a virtual route pacing window size for a route, but if that route is activated by another node, such as an NCP, the pacing window size is taken from the NCP definition, whether a specified value or a default.

The first number in the operand identifies the virtual route number; the second number identifies the transmission priority associated with the virtual route. For example, VRPWS02=(128,255) defines a minimum virtual route window size of 128 and a maximum window size of 255 for VR0, transmission priority 2.

The minimum window size must be greater than zero, and less than or equal to the maximum window size. The maximum window size must be greater than or equal to the minimum window size, and less than or equal to 255. If either value is not valid, VRPWS is ignored.

If you do not code this operand, VTAM sets the minimum window size equal to the ER length and the maximum window size equal to three times the ER length. However, if the virtual route ends in a subarea that is adjacent to VTAM, the maximum window size is set to the larger of the following two values:

- 15
- 255 minus 16*n*, where *n* is the number of explicit routes (defined or operative) that originate in the host and pass through, but do not end in, the adjacent subarea.

This increases the maximum window size for a route to a channel-attached NCP that has only a few explicit routes passing through it.

For additional information on virtual route window sizes, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Dynamic Path Update

Dynamic path update enables the VTAM operator to add or replace NCP path definitions without regenerating and reloading the NCP if the explicit route (ER) is not currently operative. In addition, a delete function for inoperative ERs is provided for both VTAM and NCP path definitions. Dynamic path update can also be used to change path definition operands, such as VR pacing window size, for inoperative routes.

VTAM distinguishes path definition sets from dynamic path update members based on the first noncomment statement in the member. If the first noncomment statement is a PATH definition statement, it is considered a path definition set. If the first noncomment statement is a VPATH or NCPPATH definition statement, it is considered a dynamic path update member.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for examples of dynamic path update and for a discussion of restrictions that apply to path definition sets and dynamic path update members.

Dynamic Path Update (NCP)

To define a dynamic path specification set for an NCP, code NCPPATH and PATH definition statements.

Table 36. Definition Statements for Dynamic Update NCP Paths

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
ncpname	NCPPATH	NETID	488	R			
name	PATH	DESTSA	487	R			
		DELETER	486				see description
		ER0–ER15	487				
		VR0–VR7	489				
		VRPWS00–72	489				

Legend:

R Required operand.

Coding Notes

NCPPATH Definition Statement: Code the NCPPATH definition statement to identify the NCP and the network to which the path update specification set applies in a dynamic path update member. This definition statement is required, and must precede the path update specifications for each network supported by the NCP.

PATH Definition Statement: Code a PATH definition statement to define routes.

Full Syntax

The full syntax for dynamic update NCP paths follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

NCPPATH:

```
►►──ncpname──NCPPATH──NETID=network_id──────────────────────────────────►◄
```

PATH:

```
►►──┬──────────┬──PATH──DESTSA=──┬─subarea_number──────────┬──────────────────►
    └─name─────┘                 └─(─subarea_number──────)─┘

►──┬──────────────────────────────────────────────────────────────────────►
   └─DELETER=──┬─explicit_route_number──────────┬──┘
               └─(─explicit_route_number──────)─┘

►──┬──────────────────────────────────────────────────────────────────────►
   │                   ┌─5000─┐   ┌─5000─┐   ┌─5000─┐   ┌─20000─┐
   └─ERn=(adjsub,tgn,──┼──────┼──,┼──────┼──,┼──────┼──,┼───────┼──)──┘
                       └─lo───┘   └─med──┘   └─hi───┘   └─tot────┘

►──┬──────────────────┬──┬──────────────────────────────────────────────►◄
   └─VRn=er_number────┘  └─VRPWSnn=─(minimum_size,maximum_size)──────────┘
```

Operand Descriptions

The following section describes the operands you can code for dynamic update NCP paths.

DELETER:

```
►►──┬──────────────────────────────────────────────────────────────────────►◄
    └─DELETER=──┬─explicit_route_number──────────┬──┘
                └─(─explicit_route_number──────)─┘
```

statement:	*PATH*
dependencies:	*see notes*

explicit_route_number is the explicit route number(s) for the route(s) to be deleted. The value of *explicit_route_number* can be ER0–ER15. Up to 16 explicit route numbers can be specified in DELETER.

The NCP deletes the specified explicit route from its routing table if the explicit route is not currently operative. If the requested explicit route is currently operative, the request is rejected and a warning message is displayed.

Notes:

1. Use extreme caution when coding this operand. Deleting one explicit route might make several routes unusable because multiple origin subareas can use the same routing table entry.
2. Use the DELETER operand only when there is no other method of correcting errors or resolving problems. If the explicit route must be redefined, you should use the replace function of dynamic path update.

DESTSA:

```
►►──DESTSA=──┬─subarea_number──────────┬──────────────────────────────────►◄
             └─(─subarea_number──────)─┘
```

statement:	*PATH*
dependencies:	*none*

Specifies the destination subarea number for the routes being defined to subarea nodes in your network. You can code more than one destination subarea as long as the route definition statement holds for each one.

ER0-ER15:

```
►►──┬──────────────────────────────────────────────────────────────────►◄
    │                   ┌─5000──┐   ┌─5000──┐   ┌─5000──┐   ┌─20000─┐
    └─ERn=(adjsub,tgn,──┼───────┼──,┼───────┼──,┼───────┼──,┼───────┼──)─┘
                        └─lo────┘   └─med───┘   └─hi────┘   └─tot───┘
```

statement:	*PATH*
dependencies:	*none*

Names the adjacent subarea and optionally specifies:

- The transmission group (TG) number for the associated explicit routes originating in the host and leading to the destination subarea.
- The NCP's transmission group flow-control thresholds. The maximum value for any threshold is 524288.

At least one ER must be specified on the PATH definition statement. You can change the ER value when the ER is inactivated or undefined.

adjsub
Specifies the subarea number (in decimal) of the adjacent subarea to which traffic should be sent from the host subarea for the explicit route being defined.

tgn
Specifies an optional transmission group number for the explicit route being defined.

lo Specifies the NCP's transmission group flow-control threshold for low priorities.

med
Specifies the NCP's transmission group flow-control threshold for medium priorities.

hi Specifies the NCP's transmission group flow-control threshold for high priorities.

tot
Specifies the NCP's transmission group flow-control threshold for the total traffic in characters.

name:

```
►►──┬──────┬──────────────────────────────────────────────────────►◄
    └name──┘
```

statement:	*PATH*
dependencies:	*none*

Provides the optional name of the PATH represented by this definition statement.

name is not checked by VTAM for validity, but its use is recommended. Its purpose is to point out the particular PATH statement if a definition error or warning message occurs during activation.

ncpname:

```
►►──ncpname──────────────────────────────────────────────────────►◄
```

statement:	*NCPPATH*
dependencies:	*must match PU name of NCP*

Specifies a 1–7 character name of the NCP to which the path update specification set applies. *ncpname* is required and must be identical to the PU name of the NCP.

If the specified NCP is not currently active or is being activated when the dynamic path update member is activated, VTAM issues a warning message and discards the path update specification set. The PATH definition statements that follow, however, are checked for validity, and warning messages are issued if there are errors.

NETID:

```
►►──NETID=network_id─────────────────────────────────────────────►◄
```

statement:	*NCPPATH*
dependencies:	*none*

Specifies a 1–8 character network identifier to which the path update specification set applies. This operand is required. If you do not code it, VTAM ignores the entire path update specification set and issues a warning message.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

If you do not define the specified NETID in the NCP, the NCP rejects the update request and sends an indicator to the SSCP that the NETID is not valid. This causes VTAM to issue a warning message.

VR0-VR7:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─VRn=er_number────┘
```

statement:	*PATH*
dependencies:	*none*
range:	*0–15*

Associates the virtual route with the explicit route.

er_number specifies the explicit route number to which the virtual route is mapped. You can change VR to ER mapping after initial activation of a path definition when the VR is inactivated or undefined.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72:

```
►►──┬──────────────────────────────────────┬──────────────────────►◄
    └─VRPWSnn=─(minimum_size,maximum_size)─┘
```

statement:	*PATH*
dependencies:	*none*

Specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition. You can change the pacing window size after initial activation of a path definition when the VR is inactivated or undefined.

The first number in the operand identifies the virtual route number; the second number identifies the transmission priority associated with the virtual route. For example, VRPWS02 = (128,255) defines a minimum virtual route window size of 128 and a maximum window size of 255 for VR0, transmission priority 2.

The minimum window size must be greater than zero and less than or equal to the maximum window size. The maximum window size must be greater than or equal to the minimum window size and less than or equal to 255. If either value is not valid, VRPWS is ignored.

VTAM sends the minimum and the maximum values specified with this operand to the NCP. If this operand is not coded, no window sizes are sent, and NCP uses already defined values, if they exist. If the window sizes are not defined for a VR when the VR is activated, NCP dynamically calculates the value at the VR activation time.

The value (0,0) has special meaning. If this value is coded, the currently specified values in the NCP, if any, are nullified and values dynamically calculated at VR activation time are used.

Dynamic Path Update (NCP)

For additional information on virtual route window sizes, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Dynamic Path Update (VTAM)

To define a dynamic path specification set for VTAM, code VPATH and PATH definition statements.

Table 37. Definition Statements for Dynamic Update VTAM Paths

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
sscpname	VPATH	NETID	492	R			
name	PATH	DESTSA	491	R			
		DELETER	491				see description
		ER0–ER15	491				
		VR0–VR7	493				
		VRPWS00–72	493				

Legend:

R Required operand.

Coding Notes

The operands on this PATH definition statement are the same as those on “VTAM Routes” on page 481, with one exception. If the SSCPNAME on the VPATH definition statement matches VTAM’s, the values of the destination subarea and adjacent subareas used in the PATH definition statement must not exceed the subarea addressing supported by the SSCP (as defined by the MXSUBNUM start option).

Code the VPATH definition statement to identify the VTAM subarea and the network to which the path update specification set applies in a dynamic path update member. This definition statement is required and must precede the path update specifications for each subarea. The VPATH definition statement is not allowed in the path definition set.

Full Syntax

The full syntax for dynamic update VTAM paths follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VPATH:

```
►►──sscpname──VPATH──NETID=network_id──────────────────────────────────────►◄
```

PATH:

```
►►──┬──────────┬──PATH──DESTSA=──┬─subarea_number──────────────────────────►
    └─name─────┘                 └─(─subarea_number──────)─┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │DELETER=──┬─explicit_route_number──────────┬──│ │ERn=──┬─(adjsub)──────────┬──│
   │          │─(─explicit_route_number──)──────│  │ │      │─(adjsub,tgn)──────│  │
   │──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │VRn=er_number──────────────────────────────────────────────────────────────────────────────────────────────────────────────────►
   │VRPWSnn=─(minimum_size,maximum_size)───────────────────────────────────────────────────────────────────────────────────────────►◄
```

Operand Descriptions

The following section describes the operands you can code for dynamic update VTAM paths.

DELETER:

```
►►──┬──────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────────►◄
    │DELETER=──┬─explicit_route_number──────────┬──│
    │          │─(─explicit_route_number──)──────│
```

statement:	*PATH*
dependencies:	*see notes*

Specifies the explicit route number(s) for the route(s) to be deleted. The value of *explicit_route_number* can be ER0–ER15. Up to 16 explicit route numbers can be specified in DELETER.

The NCP deletes the specified explicit route from its routing table if the explicit route is not currently operative. If the requested explicit route is currently operative, the request is rejected and a warning message is displayed.

Notes:

1. Use extreme caution when coding this operand. Deleting one explicit route might make several routes unusable because multiple origin subareas can use the same routing table entry.
2. Use the DELETER operand only when there is no other method of correcting errors or resolving problems. If the explicit route must be redefined, you should use the replace function of dynamic path update.

DESTSA:

```
►►──DESTSA=──┬─subarea_number──────────┬──────────────────────────────────────────────────────────────────────────────────────►◄
             │─(─subarea_number──)─────│
```

statement:	*PATH*
dependencies:	*none*

Specifies the destination subarea number for the routes being defined to subarea nodes in your network. You can code more than one destination subarea as long as the route definition statement holds for each one.

ER0-ER15:

```
►►──┬──────────────────────────────────────────────────────────────────►◄
    └ERn=──┬─(adjsub)──────┬─
           └─(adjsub,tgn)──┘
```

statement: *PATH*
dependencies: *none*

Names the adjacent subarea and optionally specifies the transmission group (TG) number for the associated explicit routes originating in the host and leading to the destination subarea.

At least one ER must be specified on the PATH definition statement. You can change the ER value when the ER is inactivated or undefined.

adjsub
Specifies the subarea number (in decimal) of the adjacent subarea to which traffic should be sent from the host subarea for the explicit route being defined.

tgn
Specifies an optional transmission group number for the explicit route being defined.

name:

```
►►──┬──────┬──────────────────────────────────────────────────────────►◄
    └name──┘
```

statement: *PATH*
dependencies: *none*

Provides the optional name of the PATH represented by this definition statement.

name is not checked by VTAM for validity, but its use is recommended. Its purpose is to point out the particular PATH statement if a definition error or warning message occurs during activation.

NETID:

```
►►──NETID=network_id──────────────────────────────────────────────────►◄
```

statement: *VPATH*
dependencies: *none*

Specifies a 1–8 character network identifier to which the path update specification set applies. This operand is required.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

The specified NETID must be identical with the NETID specified in the VTAM START command; otherwise, VTAM ignores the path update specification set.

sscpname:

```
►►──sscpname──────────────────────────────────────────────────────────►◄
```

statement:	*PATH*
dependencies:	*SSCPNAME start option*

Specifies a 1–8 character name of the VTAM subarea to which the path update specification set applies. *sscpname* is required. *sscpname* must be identical with the SSCPNAME specified in the start option currently in use; otherwise, VTAM ignores the path update specification set.

VR0-VR7:

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └─VRn=er_number────┘
```

statement:	*PATH*
dependencies:	*none*
range:	*0–15*

Associates the virtual route with the explicit route.

er_number specifies the explicit route number to which the virtual route is mapped. You can change VR to ER mapping after initial activation of a path definition when the VR is inactivated or undefined.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72:

```
►►──┬──────────────────────────────────────────┬──────────────────────►◄
    └─VRPWSnn=─(minimum_size,maximum_size)─────┘
```

statement:	*PATH*
dependencies:	*none*

Specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition. You can change the pacing window size after initial activation of a path definition when the VR is inactivated or undefined.

The first number in the operand identifies the virtual route number; the second number identifies the transmission priority associated with the virtual route. For example, VRPWS02 = (128,255) defines a minimum virtual route window size of 128 and a maximum window size of 255 for VR0, transmission priority 2.

The minimum window size must be greater than zero and less than or equal to the maximum window size. The maximum window size must be greater than or equal to the minimum window size and less than or equal to 255. If either value is not valid, VRPWS is ignored.

VTAM sends the minimum and the maximum values specified with this operand to the NCP. If this operand is not coded, no window sizes are sent, and NCP uses already defined values, if they exist. If the window sizes are not defined for a VR when the VR is activated, NCP dynamically calculates the value at the VR activation time.

The value (0,0) has special meaning. If this value is coded, the currently specified values in the NCP, if any, are nullified and values dynamically calculated at VR activation time are used.

For additional information on virtual route window sizes, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Dynamic Reconfiguration and Dynamic Change

Dynamic reconfiguration (DR) allows you to change your network configuration without deactivating the affected major node. This means, for example, that you can change resources on an NCP without deactivating the NCP.

There are several methods of dynamic reconfiguration; this book describes how to add, delete, and move resources using a DR definition statements. To implement the changes, use the VARY DRDS command to activate the DR file. For more information on the status requirements for resources before they are dynamically reconfigured and how to implement dynamic reconfiguration, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Following are the formats of the definition statements that are coded in a DR file for use with the VARY DRDS command. The statements must follow the rules described in “Format of Definition Statements” on page 9. File the completed statements in the VTAM definition library.

Add Operation

For an ADD operation, code the following definition statements:

VBUILD
ADD
PU
LU

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 497 for details on the VBUILD definition statement.

ADD Definition Statement: Code an ADD definition statement to indicate the adding of a resource.

PU and LU Definition Statements: Code one PU and LU statement for each resource to be added.

- Code the operands in the PU and LU definition statements in the same format as they appear in the appropriate major node.

- Code all desired operands from the GROUP, LINE, or PU definition statements to which you are adding the physical unit or logical unit. VTAM ignores NCP-only operands and might issue a warning message.
- Sifting takes place within the hierarchy of minor nodes being added dynamically. Values for operands coded in the original hierarchy above the hierarchy being added do not sift down to the added resources. Therefore, if you want the values coded in the original hierarchy, you must code them in the group of DR definition statements for the resources you are adding.
- The most efficient way to add a physical unit and its associated logical units is to specify them on the same ADD definition statement and not use separate ADD definition statements for the physical unit and each logical unit.

Table 38. Definition Statements for Dynamic Reconfiguration with VARY DRDS (ADD operation)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=DR	501	R			
name	ADD	TO	500	R			
		DRTYPE	499			NCP	Adding LUs to PUs only
name	PU						
name	LU						

Legend:

R Required operand.

Delete Operation

For a delete operation, code the following definition statements:

VBUILD
DELETE
PU
LU

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 497 for details on the VBUILD definition statement.

DELETE Definition Statement: Code a DELETE definition statement to indicate the deletion of a resource.

PU and LU Definition Statements: Code the PU and LU statement for the resource to be deleted. Code only the PU or LU names. You do not have to code individual PU or LU operands.

Note: When you delete a physical unit, you also delete its associated logical units. Do not code the associated LU definition statements.

Dynamic Path Update (VTAM)

Table 39. Definition Statements for Dynamic Reconfiguration with VARY DRDS (DELETE Operation)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=DR	501	R			
name	DELETE	FROM	499	R			
name	PU						
name	LU						

Legend:

R Required operand.

Move Operation

For a move operation, code the following definition statements:
VBUILD
MOVE
PU

The MOVE operation is only for physical units (and their associated logical units). To move an individual logical unit to another physical unit, use a DELETE and then a subsequent ADD. (For efficient storage management, code all DELETEs before their related ADDs.)

A DR MOVE is valid only for NCP PUs.

Notes:

1. For an SSCP to successfully take over for another SSCP that has issued a DR MOVE, the takeover SSCP must also support DR MOVE.
2. The takeover SSCP must be running VTAM V3R2 or later, and must be configured the same as the original host.

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 497 for details on the VBUILD definition statement.

MOVE Definition Statement: Code a MOVE statement to indicate the moving of a physical unit from one line to another or to change the SDLC address of a physical unit.

To change the SDLC address for a physical unit on a certain line, use the MOVE definition statement and specify the new address for the ADDR operand, keeping the FROM line and the TO line the same.

PU Definition Statement: Code a PU statement to identify the physical unit you want to move.

Code only the names of the physical units you want to move. Do not code the names of associated logical units. VTAM automatically includes the associated logical units in the move. For each physical unit, you can optionally specify a new station address and whether you want the physical unit and its logical units automatically activated after the move.

Note: Values coded for operands on the original GROUP or LINE definition statements do not sift down to operands on the new PU and LU for the resources being moved.

Table 40. Definition Statements for Dynamic Reconfiguration with VARY DRDS (MOVE Operation)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=DR	501	R			
name	MOVE	FROM	499	R			
		TO	500	R			
name	PU	ACTIVATE	498			NO	
		ADDR	499				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in a group of dynamic reconfiguration definition statements.

You can file the dynamic reconfiguration definition statements for the different types of operations as one member. If multiple operations are defined, code only one VBUILD definition statement.

Full Syntax

The full syntax for dynamic reconfiguration with VARY DRDS follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=DR──────────────────────────────────────────►◄
    └─name─────┘
```

ADD:

```
                                   ┌─,DRTYPE=NCP──────────┐
►►──┬──────────┬──ADD──TO=resource_name──┤                      ├──────────────►◄
    └─name─────┘                   └─,DRTYPE=──┬─NCP──┬──┘
                                               └─LOCAL─┘
```

DELETE:

Dynamic Path Update (VTAM)

```
►►──┬──────┬──DELETE──FROM=resource_name──────────────────────────────►◄
    └name──┘
```

MOVE:

```
►►──┬──────┬──MOVE──TO=line_name,FROM=line_name───────────────────────►◄
    └name──┘
```

PU:

```
                  ┌─ACTIVATE=NO──────┐
►►──pu_name──PU──┤                  ├──┬──────────────────────────┬──►◄
                  │                  │  └ADDR=link_station_address─┘
                  └─ACTIVATE=──┬─NO──┬┘
                               └─YES─┘
```

LU:

```
►►──lu_name──LU──────────────────────────────────────────────────────►◄
```

Operand Descriptions

The following section describes the operands you can code for dynamic reconfiguration with VARY DRDS.

ACTIVATE

```
    ┌─ACTIVATE=NO──────┐
►►──┤                  ├──────────────────────────────────────────────►◄
    │                  │
    └─ACTIVATE=──┬─NO──┬┘
                 └─YES─┘
```

statements:	*PU*
dependencies:	*none*

Specifies whether VTAM automatically activates the physical unit and its associated logical units (that have been defined with ISTATUS=ACTIVE) after the physical unit has been moved to a new line.

If you use ACTIVATE=NO, you can activate the physical unit after the move by issuing VARY ACT,ID=*pu name*.

ADDR

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─ADDR=link_station_address─┘
```

statements:	*PU*
dependencies:	*none*
range:	*X'01'–X'FE'*

Specifies a new 2-digit hexadecimal SDLC address for a physical unit being moved. The new SDLC link station address replaces the physical unit's current link station address. This operand is optional.

To move multiple physical units and change their addresses concurrently, specify a unique address for each PU under the LINE definition statement. The valid address range is X'01'–X'FE'.

DRTYPE

```
      ┌─,DRTYPE=NCP──────────┐
►►────┼──────────────────────┼──────────────────────────────────────────►◄
      │              ┌─NCP──┐│
      └─,DRTYPE=─────┼──────┼┘
                     └─LOCAL─┘
```

statements:	*ADD*
dependencies:	*Adding LUs to PUs only*

Specifies the major node you are dynamically reconfiguring. Valid values are:

NCP An NCP major node

LOCAL
A local SNA major node

FROM

```
►►──FROM=resource_name──────────────────────────────────────────────────►◄
```

statements:	*DELETE, MOVE*
dependencies:	*none*

resource_name can be any of the following:

line name
Identifies a line from which you want to delete or move a physical unit. The physical units attached to this line must be types 1, 2, or 2.1. The line must be defined as nonswitched, and can be active or inactive. This operand is required if you are deleting or moving a physical unit. The line from which you are moving the physical unit must be within the same NCP as the line to which you are moving it.

Note: If the line is under a GROUP that specifies VIRTUAL=YES, dynamic reconfiguration is not allowed for the line.

PU name
Identifies a physical unit from which you want to delete any associated logical units. The physical unit must be type 1, type 2, or type 2.1. This operand is required to identify the physical unit.

Note: If the physical unit is under a GROUP that specifies VIRTUAL=YES, dynamic reconfiguration is not allowed for the physical unit.

name

```
►►──┬──────┬──────────────────────────────────────────────────────────►◄
    └name──┘
```

statements: *VBUILD, ADD, DELETE, MOVE*
dependencies: *none*

Specifies an optional name for an individual DR statement for error-reporting purposes. If you code the name for a DR definition statement and the statement fails, VTAM includes the name in the appropriate operator message. The use of *name* is strongly recommended.

TO

```
►►──TO=resource_name──────────────────────────────────────────────────►◄
```

statements: *ADD, MOVE*
dependencies: *none*

resource_name can be one of the following:

line name
Identifies a line to which you want to add or move a physical unit. The physical units attached to this line must be type 1, 2, or 2.1. This line must be defined as nonswitched, and can be active or inactive. This operand is required if you are adding or moving a physical unit. The line to which you are moving the physical unit must be within the same NCP as the line from which you are moving it.

Note: If the line is under a GROUP that specifies VIRTUAL=YES, dynamic reconfiguration is not allowed for the line.

PU name
Identifies a physical unit to which you want to add any associated logical units. The physical unit must be type 1, type 2, or type 2.1. This operand is required to identify the physical unit.

Note: If the physical unit is under a GROUP that specifies VIRTUAL=YES, dynamic reconfiguration is not allowed for the physical unit.

Notes:

1. When coding the operands for the PU and LU definition statements, do not code NCP-only operands; otherwise, a warning message will be issued.

2. If you do not code the MAXDATA operand on the PU definition statement in a dynamic reconfiguration definition, a value of 261 bytes is used for a PU type 1 and a value of 265 is used for a PU type 2.

TYPE

```
►►──TYPE=DR──────────────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Means that this VBUILD definition statement defines a dynamic reconfiguration file to VTAM. This operand is required.

Adjacent SSCP Table

The adjacent SSCP table contains lists of SSCPs that can be in session with the VTAM SSCP and can be used to reach destination SSCPs in the same or other networks. One or more adjacent SSCP tables can be activated in each VTAM host. File the definitions in VTAMLST files (or as members in the SYS1.VTAMLST data set).

With the dynamic adjacent SSCP table function, you are not required to code adjacent SSCP tables to establish cross-domain and cross-network sessions. With this function, VTAM dynamically routes session establishment requests to all active adjacent SSCPs until the correct SSCP is found. However, if you do code adjacent SSCP tables, VTAM uses them.

Table 41. Definition Statements for ADJSSCP Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=ADJSSCP	508	R			
name	NETWORK	NETID	506				
		SORDER	507				
listname	ADJLIST		506				
sscpname	CDRM	SORDER	508				
cdrmname	ADJCDRM		505				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in an adjacent SSCP table.

NETWORK Definition Statement

Code a NETWORK definition statement to name a destination network. If you do not code a NETWORK definition statement, this adjacent SSCP table defines a default SSCP list for either a single-network or a multiple-network environment. In a multiple-network environment, you can use the NETWORK definition statement to provide a specific default SSCP list for a given destination network.

If ADJCDRM definition statements follow a NETWORK definition statement, without intervening CDRM definition statements, they also comprise a list of default SSCPs. A default SSCP list for a single network consists of a VBUILD definition statement with TYPE=ADJSSCP followed by a NETWORK definition statement and a set of ADJCDRM definition statements.

The following rules apply to the placement of the NETWORK definition statement within the adjacent SSCP table definition:

- The NETWORK definition statement can be followed by a NETWORK, CDRM, or ADJCDRM definition statement.
- Consecutive NETWORK definition statements result in the same adjacent SSCP table for each of the networks.
- If a statement that is not valid is encountered within the adjacent SSCP table definition, that statement and any subsequent statements are rejected until another NETWORK definition statement or the end of the definition is encountered.

Note: An ADJSSCP table defined with a NETWORK definition statement of the host network is used only when the destination resource is known to reside in the host network. A default table is coded without a NETWORK definition statement.

CDRM Definition Statement

Code a CDRM definition statement to define the adjacent SSCP table as a CDRM. One or more CDRM definition statements can follow a NETWORK or ADJCDRM definition statement.

In an SNA interconnected network, the CDRM definition statement names a destination SSCP in the network identified on the preceding NETWORK definition statement.

The following rules apply to the placement of the CDRM statement within the adjacent SSCP table definition:

- A CDRM definition statement can be followed by another CDRM definition statement.
- One or more CDRM definition statements must be followed by an ADJCDRM definition statement.

The CDRM definition statements within the adjacent SSCP table have no required operands, although any operands that are valid on the CDRM definition statement in a CDRM major node are accepted. These operands are ignored when the CDRM definition statement is part of an adjacent SSCP table.

Note: You can define adjacent SSCP tables to reach a new interconnected network by adding only a few statements to the existing CDRM definition of that network.

ADJLIST Definition Statement

Code an ADJLIST definition statement to define the name of a list of adjacent SSCPs. The list of adjacent SSCPs is created from the ADJCDRM definition statements which follow one or more ADJLIST definition statements.

An ADJLIST statement must be immediately followed by either:
- One or more ADJCDRM definition statements
- One or more ADJLIST definition statements

Coding any other definition signifies the end of the ADJLIST definitions. Once the first ADJCDRM definition statement has been encountered immediately following a ADJLIST definition statement, the next occurrence of an ADJLIST definition statement indicates the start of a new ADJLIST. The following example shows two lists that have been coded. LIST1 contains GW1 and GW2. LIST2 contains GW1, GW2, and GW3.

```
       VBUILD  TYPE=ADJSSCP
LIST1  ADJLIST
GW1    ADJCDRM
GW2    ADJCDRM
LIST2  ADJLIST
GW1    ADJCDRM
GW2    ADJCDRM
GW3    ADJCDRM
```

Multiple lists can be set up to contain identical ADJCDRMs without duplication of effort. Code all ADJLIST definition statements immediately followed by all the ADJCDRMs for which there would be an entry in each list. For example, using the following definition statements GW1, GW2, and GW3 reside in both LIST1 and LIST2.

```
LIST1  ADJLIST
LIST2  ADJLIST
GW1    ADJCDRM
GW2    ADJCDRM
GW3    ADJCDRM
```

Note: If an ADJLIST definition statement follows a NETWORK, CDRM, or ADJCDRM definition statement, those definitions are considered ended. This might result in a problem if the ADJLIST definition statement is placed incorrectly and the previous definition has not been completed when encountered.

ADJCDRM Definition Statement

Code consecutive ADJCDRM definition statements to create a list of default SSCPs.

Consecutive ADJCDRM definition statements following the CDRM definition statements give a list of adjacent SSCP names for the preceding destination SSCPs identified on the CDRM definition statements.

Full Syntax

The full syntax for the adjacent SSCP table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=ADJSSCP──────────────────────────────────────►◄
    └name──┘
```

NETWORK

```
                                                        (1)
                                          ┌─SORDER=STARTOPT──────┐
►►──┬──────┬──NETWORK──┬──────────────────┼──────────────────────┼──────►◄
    └name──┘           └NETID=network_id─┘ └─SORDER=──┬─APPNFRST─┬─┘
                                                       ├─APPN────┤
                                                       ├─ADJSSCP─┤
                                                       ├─SUBAREA─┤
                                                       └─STARTOPT─┘
```

Notes:

1 For CDRM, SORDER defaults to the value coded on the preceding NETWORK statement.

ADJLIST

```
►►──listname──ADJLIST──────────────────────────────────────────────────►◄
```

CDRM

```
                                  (1)
                     ┌─SORDER=STARTOPT──────┐
►►──sscpname──CDRM──┼──────────────────────┼──────────────────────────►◄
                     └─SORDER=──┬─APPNFRST─┬─┘
                                ├─APPN────┤
                                ├─ADJSSCP─┤
                                ├─SUBAREA─┤
                                └─STARTOPT─┘
```

Notes:

1 For CDRM, SORDER defaults to the value coded on the preceding NETWORK statement.

ADJCDRM

```
►►──cdrmname──ADJCDRM──────────────────────────────────────────────────►◄
```

Operand Descriptions

The following section describes the operands you can code for the adjacent SSCP table.

cdrmname

```
►►──cdrmname──────────────────────────────────────────────────────────►◄
```

statements: *ADJCDRM*
dependencies: *SORDER and SSCPORD start options*

The name of an adjacent SSCP as known to VTAM. It must be the same as the name on the CDRM definition statement that defines the SSCP to VTAM.

Each ADJCDRM definition statement should have a unique *cdrmname*. Otherwise, an error message is issued. Furthermore, in an SNA interconnected network, the CDRM names must be unique among SSCPs that participate in cross-network sessions.

The following rules apply to the placement of the ADJCDRM definition statement within the adjacent SSCP table definition:

- The ADJCDRM definition statement can be followed by an ADJCDRM, ADJLIST, CDRM, or NETWORK definition statement (or nothing, if it is the end of the table).
- If an ADJCDRM definition statement immediately follows a NETWORK definition statement, it is the default adjacent SSCP table for the specified network.
- If an ADJCDRM definition statement immediately follows the VBUILD definition statement or a NETWORK definition statement without NETID specified, it is the default adjacent SSCP table for all networks.

You can specify ISTAPNCP, instead of an SSCP name, for *cdrmname*. ISTAPNCP is a generic representation for an APPN CDRM. It can provide you with additional control over the order in which the network is searched.

If you code the SORDER=ADJSSCP start option and code ISTAPNCP in any or all of the user-defined adjacent SSCP tables, VTAM does not search the APPN network until it encounters ISTAPNCP in the adjacent SSCP table.

That is, if you code ISTAPNCP as the fourth CDRM in the user-defined adjacent SSCP table, VTAM does not search the APPN network until after it has searched unsuccessfully in the first three CDRMs in the table. If you code ISTAPNCP first in the adjacent SSCP table, VTAM searches the APPN network immediately.

Notes:

1. If SSCPORD=PRIORITY and SORDER=ADJSSCP, and you do not code ISTAPNCP in the user-defined adjacent SSCP table, VTAM does not add ISTAPNCP to the adjacent SSCP routing table, and VTAM does not search the APPN network unless a previous request for this DLU has searched the APPN network. (That is, there is an ISTAPNCP entry in the ISTADSST table queued to the DLU RDTE.)
2. If you code the SORDER=APPN, SORDER=SUBAREA, or SORDER=APPNFRST start options (or let it default to APPN), VTAM ignores the ISTAPNCP entry in the user-defined adjacent SSCP table, since VTAM places ISTAPNCP in the appropriate place in the adjacent SSCP routing table.
3. If you code the SSCPORD=PRIORITY start option (or let it default to PRIORITY), and either SORDER=APPNFRST, SORDER=APPN, or

SORDER=SUBAREA, VTAM searches the adjacent SSCP table in the following order. ISTAPNCP is added in one of the following places:

- Before step 3a, if SORDER=APPNFRST
- After step 3b, if SORDER=APPN
- After step 3d, if SORDER=SUBAREA

a. The CDRM that owns the resource (if known)
b. The SSCPs for which the most recent session-initiation attempt succeeded
c. The SSCPs for which no session-initiation attempt has been made
d. The SSCPs for which the last session-initiation attempt failed.

4. If you code the SSCPORD=DEFINED start option, VTAM searches the adjacent SSCP table in the same order that SSCPs are listed, with the exception that the CDRM that owns the resource (if known) is searched first.

listname

```
►►──listname──────────────────────────────────────────────────────────►◄
```

statements: *ADJLIST*
dependencies: *none*

Defines the name of a list of adjacent SSCPs. This name is used by the ADJLIST operand on the CDRSC definition statement to specify which adjacent SSCPs should be used for all session setup requests for that resource.

The following names are reserved adjacent list names and may not be used for *listname*:

ALL
DYNAMIC
DEFAULT
RESOURCE

name

```
►►──┬──────┬──────────────────────────────────────────────────────────►◄
    └name──┘
```

statements: *VBUILD, NETWORK*
dependencies: *none*

Provides the optional name of the adjacent SSCP table represented by this definition statement or identifies this NETWORK definition statement. To avoid confusion, use the same value for both *name* and the NETID operand.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────────►◄
    └NETID=network_id──┘
```

statements: *NETWORK*

dependencies: *none*

Specifies the optional 1–8 character identifier of the destination network. Omitting NETID defines a default SSCP list for all networks.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

SORDER

```
                    (1)
   ┌SORDER=STARTOPT──────┐
►►─┤                     ├──────────────────────────────────────────►◄
   └SORDER=──┬APPNFRST─┬─┘
             ├APPN────┤
             ├ADJSSCP─┤
             ├SUBAREA─┤
             └STARTOPT┘
```

Notes:

1 For CDRM, SORDER defaults to the value coded on the preceding NETWORK statement.

statements: *NETWORK , CDRM*
dependencies: *none*

Note: When coded on a NETWORK statement, the SORDER value ″sifts″ down to all CDRM statements under that NETWORK statement (unless SORDER is overridden on one or more of the CDRM statements).

SORDER=APPNFRST
Specifies that VTAM should always try searching the APPN network first, even if the ″owning″ CDRM was defined or learned.

SORDER=APPN
Specifies that VTAM should try searching the APPN network first, unless the ″owning″ CDRM was defined or learned.

SORDER=ADJSSCP
Specifies that VTAM should only search the APPN network if ISTAPNCP has been explicitly coded within the ADJSSCP table that was chosen for this search/session.

SORDER=SUBAREA
Specifies that VTAM should always try searching the APPN network last (after trying all subarea ADJSSCPs).

SORDER=STARTOPT
Specifies that VTAM should use the ″then current″ value of the SORDER start option for this ADJSSCP table. This value can be used on CDRM statements to override the ″sift down″ value from the preceding NETWORK statement. When coded on a NETWORK statement, this value works the same way as when SORDER= is not coded on the NETWORK statement.

sscpname

```
►►──sscpname──────────────────────────────────────────────────────────►◄
```

statements:	*CDRM*
dependencies:	*must match SSCPNAME start option*

Provides the name of the SSCP. It must match the name assigned to that SSCP in its own network by the SSCPNAME start option.

The *sscpname* field on each CDRM definition statement names a destination SSCP in the network identified by the preceding NETWORK definition statement, or, if there is no NETWORK definition statement, in the network where the table is defined.

TYPE

```
►►──TYPE=ADJSSCP──────────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Specifies that this VBUILD definition statement defines an adjacent SSCP table.

Adjacent Cluster Routing Definitions

The adjacent cluster (ADJCLUST) routing list allows you to define which adjacent APPN subnetworks VTAM should search. For each destination network, you can define a table that specifies the control point (CP) name of the adjacent nodes to which the search is sent.

Each routing list contains one or more network identifiers, followed by the CPNAMEs of the nodes that should be searched when VTAM receives a request to search for a resource with that NETID.

Figure 1 contains the following two routing lists:

- A default routing list to be used when one of the following conditions is met.
  - A non-network qualified request is received.
  - A network qualified request is received and the NETID specified is not defined in any NETWORK statement.
- A list (NETAC) to be used when NETA or NETC is the network identifier specified on a request.

```
ADJCLUST VBUILD    TYPE=ADJCLUST
*******************************************************************
* Default network ID                                              *
*******************************************************************
DEFAULT  NETWORK   SNVC=4,                allow depth of 4 networks   x
                   BNDYN=LIMITED          allow limited dynamics
NODE2A   NEXTCP    CPNAME=NETA.NODE2A     route to NODE2A
NODE2B   NEXTCP    CPNAME=NETB.NODE2B     route to NODE2B
*******************************************************************
* Routing for NETID=NETA and NETID=NETC                           *
*******************************************************************
NETAC    NETWORK   NETID=(NETA,NETC),                                x
                   BNDYN=NONE,            do not allow dynamics       x
                   SNVC=4                 allow depth of 4 subnets    x
NODE2A   NEXTCP    CPNAME=NETA.NODE2A     route to NODE2A
NODE2C   NEXTCP    CPNAME=NETC.NODE2C     route to NODE2C
```

Figure 1. Adjacent Cluster Routing Definitions

Table 42. Definition Statements for ADJCLUST Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=ADJCLUST	513	R			
name	NETWORK	BNDYN	511			BNDYN start option value	BN=YES
		NETID	512				
name	NEXTCP	CPNAME	511				*this node's cpname*
		SNVC	512		N	SNVC start option	

Legend:

R Required operand.

N Code this operand on the higher-level NETWORK definition statement to take advantage of the sift effect, or code it on this definition statement.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in an adjacent cluster definition list.

NETWORK Definition Statement

Code a NETWORK definition statement for each NETID or group of NETIDs that you want to define routing.

NEXTCP Definition Statement

Code a NEXTCP definition statement for each node that is to be searched.

Full Syntax

The full syntax for the adjacent cluster routing definitions follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=ADJCLUST──────────────────────────────────►◄
    └name──┘
```

NETWORK

```
                          ┌─BNDYN=BNDYN_start_option_value─┐
►►──┬──────┬──NETWORK──┬──┴────────────────────────────────┴──────────────►
   └name──┘           └─BNDYN=──┬──────┬──┬──────────────────────────┬──┘
                                ├─FULL───┤  │                          │
                                ├─LIMITED┤  │                          │
                                └─NONE───┘  │                          │

►──┬──────────────────────────────────────┬──────────────────────────────►◄
   └─NETID=──┬─network_id──────────────┬──┘
             └─(─network_id──────────)─┘
```

NEXTCP

```
                          ┌─CPNAME=this_node's_cpname──────────────┐
►►──┬──────┬──NEXTCP──┬──┴────────────────────────────────────────┴──────►
   └name──┘         └─CPNAME=──┬─control_point_name──────────┬──┘
                               └─netid_control_point_name────┘

          ┌─value_coded_on_snvc_start_option─┐
►──┬─SNVC=─┴──────────────────────────────────┴──┬──────────────────────►◄
   │                                            │
   │          ┌─value_coded_on_snvc_start_option─┐
   └─SNVC=────┴──────────────────────────────────┴──┘
             └─subnetwork_visit_count──────────────┘
```

Operand Descriptions

The following section describes the operands you can code for the adjacent cluster routing definition statements.

BNDYN

```
   ┌─BNDYN=BNDYN_start_option_value─┐
►►─┤                                ├──────────────────────────────────────►◄
   └─BNDYN=──┬─FULL────┬──────────┘
             ├─LIMITED─┤
             └─NONE────┘
```

statements: *NETWORK*
dependencies: *BN=YES start option coded*

Specifies how VTAM will add nodes dynamically to the routing list.

BNDYN=NONE
No nodes are added except those explicitly identified by NEXTCP definition statements.

BNDYN=LIMITED
All active border nodes in the native subnetwork as well as active adjacent nodes in nonnative subnetworks attached to this node are automatically added to the routing list if either of the following conditions is met:

- BNs and nonnative NNs that VTAM learns about whose network ID matches the network ID of the destination LU.
- BN or nonnative NN that have previously sent a Locate that carried the network ID of the destination LU.

BNDYN=FULL
All active border nodes in the native subnetwork as well as active adjacent nodes in nonnative subnetworks attached to this node are automatically added to the routing list.

CPNAME

```
   ┌─CPNAME=this_node's_cpname──────────┐
►►─┤                                    ├──────────────────────────────────────►◄
   └─CPNAME=──┬─control_point_name──────┬──┘
              └─netid_control_point_name─┘
```

statements: *NEXTCP*
dependencies: *none*

Specifies the CP name of the node to which the search request is to be forwarded.

The CP name may be network qualified. If you specify a *control_point_name*, but not a *netid*, VTAM uses this node's NETID. If you do not code CPNAME, VTAM uses this node's network qualified CP name.

The CP names of the following nodes can be listed when defining ADJCLUST definitions:

- Extended border nodes in this node's topology subnetwork, which own subnetwork boundaries with other APPN subnetworks.
- Nodes in adjacent nonnative subnetwork with which this node has a subnetwork boundary.

- This node's CPNAME which indicates that this node's native subnetwork should be searched.

name

```
►►──┬──────────┬──────────────────────────────────────────────────────►◄
    └─name─────┘
```

statements: *VBUILD, NETWORK, NEXTCP*
dependencies: *none*

Assigns a name to the definition statement.

To avoid confusion, use the same value for *name* and the following:
- CPNAME when the NEXTCP definition statement is coded
- NETID when the NETID definition statement is coded

NETID

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─NETID=──┬─network_id─────────┬┘
              └─(─network_id──────)┘
```

statements: *NETWORK*
dependencies: *none*

Specifies the optional 1–8 character identifier of the destination network.

If you do not code NETID at all, this NETWORK definition statement represents an adjacent cluster routing definition to be used for unknown or unlisted NETIDs (those not explicitly coded on other NETWORK statements).

Do not code the same NETID multiple times within a routing list. If a duplicate value is specified, only the last one listed is used.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

SNVC

```
                ┌─value_coded_on_snvc_start_option─┐
    ┌─SNVC=─────┴──────────────────────────────────┴┐
►►──┤                                               ├──────────────────►◄
    │                ┌─value_coded_on_snvc_start_option─┐
    └─SNVC=──────────┴──────────────────────────────────┴┘
                     └─subnetwork_visit_count────────────┘
```

statements: *NETWORK, NEXTCP*

dependencies: *none*
range: *1–255*

Defines a maximum number of subnetworks this border node will search when looking for a resource. If no value is coded for SNVC, the value specified on the SNVC start option is used.

Notes:

1. If SNVC=1 is coded, the search scope is restricted to the originating subnetwork. A value of SNVC=2 is required to allow a request to traverse one subnetwork boundary.
2. The maximum subnetwork count can be modified downwards if a node along the path has a lower SNVC count. However, the received SNVC is never modified upwards.

TYPE

```
►►──TYPE=ADJCLUST──────────────────────────────────────────────────────►◄
```

statement: *VBUILD*
dependencies: *none*

Indicates that this VBUILD definition statement defines a set of adjacent cluster definitions to VTAM.

APPN Transmission Group Profile

Define a transmission group (TG) profile to specify a set of characteristics for a transmission group. The TG profile is not a resource, but describes the TG characteristics of a link.

An adjacent link station (PU) calls a TG profile. When an adjacent link station is activated, VTAM attempts to locate the TG profile. If the TG profile cannot be found, default values are used for the TG characteristics.

A newer TG profile with the same name as a previously defined TG profile updates the older TG profile. The update does not affect existing connections, but subsequent calls to the updated TG profile receive the new information.

TGP Definition Statement

Code one TGP definition statement and optional operands for each TG profile.

IBM-Supplied Transmission Group Profiles

The IBM-supplied transmission group profiles are shipped with the name IBMTGPS in SYS1.ASAMPLIB. Copy IBMTGPS into SYS1.VTAMLST at VTAM installation.

Note: Unlike the COSAPPN member, the IBMTGPS member is not automatically activated. If you want to activate it, use the VARY ACT command, or place the IBMTGPS member in the configuration list to automatically activate it at VTAM initialization.

Adjacent Cluster Routing Definitions

See “APPN Transmission Group (TG) Profile Definitions (IBMTGPS)” on page 824 for samples of the profiles, and refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more details.

Table 43. Definition Statement for Transmission Group Profile (TGP)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	TGP	CAPACITY	515			8K	
		COSTBYTE	518			0	
		COSTTIME	519			0	
		PDELAY	519			TERRESTR	
		SECURITY	520			UNSECURE	
		UPARM1	520			128	
		UPARM2	521			128	
		UPARM3	521			128	

Legend:

R Required operand.

Full Syntax

The full syntax for the transmission group profile follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

TGP

```
►►──name──TGP──┬─CAPACITY=8K──────────────────┬──┬─COSTBYTE=0──────────────────┬──►
               └─CAPACITY=number_of_bits──────┘  └─COSTBYTE=cost_per_byte──────┘

►──┬─COSTTIME=0──────────────────────┬──┬─PDELAY=TERRESTR──────────┬──►
   └─COSTTIME=cost_per_unit_time─────┘  │                          │
                                        │                   (1)    │
                                        ├─PDELAY=NEGLIGIB──────────┤
                                        │                          │
                                        └─PDELAY=┬─TERRESTR──┬──────┘
                                                 ├─LONG──────┤
                                                 ├─NEGLIGIB──┤
                                                 └─PACKET────┘

►──┬─SECURITY=UNSECURE──────────────┬──┬─UPARM1=128──────────────────────────┬──►
   │                                │  └─UPARM1=user-defined_operand_value────┘
   └─SECURITY=┬─UNSECURE──┬──────────┘
              ├─PUBLIC────┤
              ├─UNDERGRO──┤
              ├─SECURE────┤
              ├─GUARDED───┤
              ├─ENCRYPT───┤
              └─SHIELDED──┘
```

```
►─┬─UPARM2=128──────────────────────────┬─┬─UPARM3=128──────────────────────────┬─►◄
  │                                     │ │                                     │
  └─UPARM2=user-defined_operand_value─┘ └─UPARM3=user-defined_operand_value─┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

Operand Descriptions

The following section describes the operands you can code for the APPN transmission group profile.

CAPACITY

```
►►─┬─CAPACITY=8K──────────────┬──────────────────────────────────────────►◄
   │                          │
   └─CAPACITY=number_of_bits─┘
```

range: *1K–1000M*

Specifies the effective capacity of the link that comprises the TG. Specify the value in either Kb per second (for example, 100K) or Mb per second (for example, 100M). This number approximates the bits per second that the link can transmit (the transmission rate of the link, times the maximum load factor expressed as a percentage).

Because the value for CAPACITY is represented as a single byte, the precision of the specified number of bits might be lost. For example, numbers that are close (such as 100K and 101K) can be interpreted by VTAM, and displayed as the same value.

See Table 44 for a list of CAPACITY values you can specify and their corresponding values when displayed. The values appearing in the VALUE CODED column are the values you can specify for CAPACITY. The values in the D NET,TGP column are the hexadecimal values that VTAM displays for the specified value when you issue D NET,TGP. The values in the D NET,TOPO column are the values that VTAM displays for the specified value when you issue D NET,TOPO. So, if you specify 80K for CAPACITY: D NET,TGP displays X'48', and D NET,TOPO displays 77K.

Table 44. CAPACITY Values When Displayed

VALUE CODED	D NET, TGP	D NET, TOPO
1K	14	1K
2K	1C	2K
3K	22	3K
4K	25	4K
5K	28	5K
6K	2A	6K
7K	2B	7K
8K	2D	8K

Table 44. CAPACITY Values When Displayed (continued)

VALUE CODED	D NET, TGP	D NET, TOPO
9K	2F	9K
10K	30	10K
11K	31	11K
12–13K	32	12K
14K	33	14K
15K	34	15K
16K	35	16k
17K	36	17K
18–19K	37	18K
20–21K	38	20K
22–23K	39	22K
24–26K	3A	24K
27–28K	3B	27K
29–31K	3C	29K
32–33K	3D	32K
34–35K	3E	34K
36–38K	3F	36K
39–43K	40	39K
44–47K	41	44K
48–52K	42	48K
53–57K	43	53K
58–62K	44	58K
63–67K	45	63K
68–71K	46	68K
72–76K	47	72K
77–86K	48	77K
87–95K	49	87K
96–105K	4A	96K
106–115K	4B	106K
116–124K	4C	116K
125–134K	4D	125K
135–143K	4E	135K
144–153K	4F	144K
154–172K	50	154K
173–191K	51	173K
192–211K	52	192K
212–230K	53	212K
231–249K	54	231K
250–268K	55	250K
269–287K	56	269K

Table 44. CAPACITY Values When Displayed (continued)

VALUE CODED	D NET, TGP	D NET, TOPO
288–307K	57	288K
308–345K	58	308K
346–383K	59	346K
384–422K	5A	384K
423–460K	5B	423K
461–499K	5C	461K
500–537K	5D	500K
538–575K	5E	538K
576–614K	5F	576K
615–691K	60	615K
692–767K	61	692K
768–844K	62	768K
845–921K	63	845K
922–998K	64	922K
999–1M	65	999K
2M	6D	2M
3M	71	3M
4M	75	4M
5M	78	5M
6M	79	6M
7M	7B	7M
8M	7D	8M
9M	7E	9M
10–11M	80	10M
12M	81	12M
13M	82	13M
14M	83	14M
15M	84	15M
16–17M	85	16M
18M	86	18M
19M	87	19M
20–22M	88	20M
23–24M	89	23M
25–27M	8A	25M
28–29M	8B	28M
30–31M	8C	30M
32–34M	8D	32M
35–36M	8E	35M
37–39M	8F	37M
40–44M	90	40M

Table 44. CAPACITY Values When Displayed (continued)

VALUE CODED	D NET, TGP	D NET, TOPO
45–49M	91	45M
50–54M	92	50M
55–58M	93	55M
59–63M	94	59M
64–68M	95	64M
69–73M	96	69M
74–78M	97	74M
79–88M	98	79M
89–98M	99	89M
99–108M	9A	99M
109–117M	9B	109M
118–127M	9C	118M
128–137M	9D	128M
138–147M	9E	138M
148–157M	9F	148M
158–176M	A0	158M
177–196M	A1	177M
197–216M	A2	197M
217–235M	A3	217M
236–255M	A4	236M
256–275M	A5	256M
276–294M	A6	276M
295–314M	A7	295M
315–353M	A8	315M
354–393M	A9	354M
394–432M	AA	393M
433–471M	AB	433M
472–511M	AC	472M
512–550M	AD	511M
551–589M	AE	551M
590–629M	AF	590M
630–707M	B0	630M
708–786M	B1	708M
787–865M	B2	787M
866–943M	B3	865M
944–1000M	B4	944M

COSTBYTE

```
   ┌─COSTBYTE=0──────────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   └─COSTBYTE=cost_per_byte──┘
```

range: *0–255*

Specifies a cost-per-byte-transmitted to be associated with the TG, where 0 is the least expensive cost per byte and 255 is the most expensive.

COSTTIME

```
   ┌─COSTTIME=0──────────────────┐
►►─┤                             ├──────────────────────────────────────►◄
   └─COSTTIME=cost_per_unit_time─┘
```

range: *0–255*

Specifies a cost-per-unit-of-time to be associated with the TG, where 0 is the least expensive cost per unit of time and 255 is the most expensive.

name

```
►►─name──────────────────────────────────────────────────────────────────►◄
```

Specifies the required name of this TG profile.

PDELAY

```
   ┌─PDELAY=TERRESTR─────────┐
   │                   (1)   │
   ├─PDELAY=NEGLIGIB─────────┤
►►─┤                         ├──────────────────────────────────────────►◄
   │           ┌─TERRESTR─┐  │
   └─PDELAY=───┤          ├──┘
               ├─LONG─────┤
               ├─NEGLIGIB─┤
               └─PACKET───┘
```

Notes:

1 NEGLIGIB is the default if the TRLE operand is coded.

Specifies the maximum propagation delay of the link for the transmission group. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

PDELAY=LONG
Indicates a satellite delay (greater than 245.76 milliseconds).

PDELAY=NEGLIGIB
Indicates a local area network delay (less than 0.48 milliseconds).

PDELAY=PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=TERRESTR
Indicates telephone network delay (between 0.48 and 49.152 milliseconds).

SECURITY

```
   ┌─SECURITY=UNSECURE──────────┐
►►─┼────────────────────────────┼──────────────────────────────────────────────►◄
   │                ┌─UNSECURE─┐│
   └─SECURITY=──────┼──────────┼┘
                    ├─PUBLIC───┤
                    ├─UNDERGRO─┤
                    ├─SECURE───┤
                    ├─GUARDED──┤
                    ├─ENCRYPT──┤
                    └─SHIELDED─┘
```

Specifies the security level of the transmission group. The following options range from the least secure level to the most secure level.

SECURITY=UNSECURE
Specifies no security level

SECURITY=PUBLIC
Specifies a public switched network

SECURITY=UNDERGRO
Specifies an underground cable, not guarded

SECURITY=SECURE
Specifies a secure conduit, not guarded

SECURITY=GUARDED
Specifies a guarded conduit, physical only

SECURITY=ENCRYPT
Specifies link encryption

SECURITY=SHIELDED
Specifies a guarded conduit, physical and radiation shielded

UPARM1

```
   ┌─UPARM1=128──────────────────────────┐
►►─┼─────────────────────────────────────┼──────────────────────────────────────►◄
   └─UPARM1=user-defined_operand_value──┘
```

range: *0–255*

Specifies a user-defined operand value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2

```
►►─┬─UPARM2=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM2=user-defined_operand_value───┘
```

range: *0–255*

Specifies a user-defined operand value.

UPARM3

```
►►─┬─UPARM3=128──────────────────────────┬──────────────────────────────────►◄
   └─UPARM3=user-defined_operand_value───┘
```

range: *0–255*

Specifies a user-defined operand value.

APPN-to-Subarea Class-of-Service Mapping Table

The APPN-to-subarea (APPNTOSA) class-of-service (COS) mapping table enables you to define how APPN COS names are mapped to subarea COS names for a VR-based TG portion of an HPR route. Subarea COS names are needed for a VR-based TG because VR-based TGs use subarea virtual routes. The subarea COS name used to find the appropriate VRs to set up an HPR route is determined as follows:

1. If an APPNTOSA COS mapping table is active and a specific subarea COS is found for a given APPN COS, that subarea COS is used. Since a blank LOGMODE name is valid in subarea, use blanks after COS= to map to it. A blank logmode name is NOT valid in APPN. (If the blank LOGMODE name is the default, code as “COS=,DEFAULT=YES”).
2. If an APPNTOSA COS mapping table is active, a specific subarea COS is not found for a given APPN COS, and a default subarea COS is coded, the default subarea COS is used.
3. If an APPNTOSA COS mapping table is active, a specific subarea COS is not found for a given APPN COS, and a default subarea COS is not coded, the ISTVTCOS class-of-service is used if coded.
4. If no APPNTOSA COS mapping table is active, the ISTVTCOS class-of-service is used if coded.
5. If none of the above conditions are met, route setup fails.

Figure 2 on page 522 illustrates definition of an APPNTOSA COS mapping table.

APPN-to-Subarea Class-of-Service Mapping Table

```
TABLE1   VBUILD    TYPE=APPNTOSA
#CONNECT MAPSTO    COS=COSAPPL1,DEFAULT=YES
#BATCH   MAPSTO    COS=COSAPPL2
#BATCHSC MAPSTO    COS=COSAPPL3
#INTER   MAPSTO    COS=
SNASVCMG MAPSTO    COS=COSAPPL5
#INTERSC MAPSTO    COS=ISTCOSDF
```

Figure 2. APPN-to-Subarea COS Mapping Table Example

Table 45. Definition Statements for APPNTOSA Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=APPNTOSA	523	R			
appn_cos	MAPSTO	COS	523	R			
		DEFAULT	523				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in an APPN-to-subarea COS mapping table.

MAPSTO Definition Statement

Code a MAPSTO definition statement to define an APPN-to-subarea COS mapping.

Full Syntax

The full syntax for the APPN-to-subarea COS mapping table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=APPNTOSA──────────────────────────────────►◄
    └name──┘
```

MAPSTO

```
►►──appn_cos──MAPSTO──COS=subarea_cos──┬──────────────┬──────────────►◄
                                       └DEFAULT=YES───┘
```

Operand Descriptions

The following section describes the operands you can code for the APPN-to-subarea COS mapping table.

appn_cos

```
►►──appn_cos──────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a required 1–8 character name identifying an APPN COS name.

COS

```
►►──COS=subarea_cos───────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a 0–8 character name identifying a subarea COS name. (If the subarea COS name is blanks, just code COS=).

DEFAULT=YES

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    └─DEFAULT=YES──┘
```

statements: *MAPSTO*
dependencies: *none*

Specifies that this entry is the default subarea COS for this table.

name

```
►►──┬──────────┬──────────────────────────────────────────────────►◄
    └─name─────┘
```

statements: *VBUILD*
dependencies: *none*

Assigns a name to the APPN-to-subarea COS mapping table.

TYPE

```
►►──TYPE=APPNTOSA─────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Means that this VBUILD definition statement defines an APPN-to-subarea COS mapping table.

Subarea-to-APPN Class-of-Service Mapping Table

The subarea-to-APPN (SATOAPPN) class-of-service (COS) mapping table enables you to define how subarea COS names are mapped to APPN COS names when a session goes from the subarea part of the network to the APPN part of the network.

VTAM picks an APPN COS as follows:

1. APPNCOS is first picked up from the requested LOGMODE entry, if it exists. A blank Subarea LOGMODE name is valid.
2. The new table, if present, is searched for an APPN mapping of the subarea COS.
3. If a subarea COS to APPN COS mapping is in the table, and it is different from the APPN COS found in the logon mode table entry, or a DEFAULT entry is coded in the table, the SATOAPPN COS table entry is used for this session.
4. If no mapping exists, the APPNCOS from the logon mode table is used, or the APPNCOS default is used if no APPNCOS was in the LOGMODE entry of the logon mode table.

Figure 3 illustrates definition of a SATOAPPN COS mapping table.

```
TABLE2   VBUILD    TYPE=SATOAPPN
COSAPPL1 MAPSTO    COS=#CONNECT,DEFAULT=YES
COSAPPL2 MAPSTO    COS=#BATCH
COSAPPL3 MAPSTO    COS=#BATCHSC
         MAPSTO    COS=#INTER
COSAPPL5 MAPSTO    COS=SNASVCMG
COSAPPL6 MAPSTO    COS=#INTERSC
```

Figure 3. Subarea-to-APPN COS Mapping Table Example

Table 46. Definition Statements for SATOAPPN Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=SATOAPPN	526	R			
sa_cos	MAPSTO	COS	525	R			
		DEFAULT	525				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in a subarea-to-APPN COS mapping table.

MAPSTO Definition Statement

Code a MAPSTO definition statement to define a subarea-to-APPN COS mapping.

Full Syntax

The full syntax for the subarea-to-APPN COS mapping table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=SATOAPPN──────────────────────────────────►◄
    └─name─────┘
```

MAPSTO

```
►►──sa_cos──MAPSTO──COS=appn_cos──┬──────────────┬──────────────────────►◄
                                  └─DEFAULT=YES──┘
```

Operand Descriptions

The following section describes the operands you can code for the subarea-to-APPN COS mapping table.

sa_cos

```
►►──sa_cos──────────────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a required 0–8 character name identifying a subarea COS name. (A blank subarea COS name is valid).

COS

```
►►──COS=appn_cos────────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a 1–8 character name identifying an APPN COS name.

DEFAULT=YES

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └─DEFAULT=YES──┘
```

statements: *MAPSTO*
dependencies: *none*

Specifies that this entry is the default APPN COS for this table.

name

```
►►──┬──────┬──────────────────────────────────────────────────────►◄
    └name──┘
```

statements: *VBUILD*
dependencies: *none*

Assigns a name to the subarea-to-APPN COS mapping table.

TYPE

```
►►──TYPE=SATOAPPN──────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Means that this VBUILD definition statement defines an subarea-to-APPN COS mapping table.

Border Node COS Mapping Definitions

The border node class-of-service (COS) mapping definitions (BNCOSMAP) enable you to define how the COS name from an adjacent APPN network (that is, a nonnative COS name) should be mapped to the local network COS name (native COS name). The border node COS mapping definitions enables each subnetwork to maintain its own COS names.

You can code the COS mapping definitions to map a nonnative COS name to a local COS name for a specific NETID. You can also define code definitions to be used if you have not defined an explicit table for the NETID.

If a table is coded, the nonnative COS name is located in the table and mapped to the corresponding native COS name. If an entry for the nonnative COS name is not found, the COS name in the adjacent subnetwork is used provided that the received COS is defined for this node by the APPN class-of-service definitions. If the received COS name is not defined, then the COS name coded on the APPNCOS start option will be used.

Figure 4 defines an explicit COS mapping table for NETWORKA, NETWORKB, and NETWORKC. The same mapping table is used for both NETWORKB and NETWORKC. The example also defines a default COS mapping table, DEFAULT, to be used if an explicit table cannot be found.

```
TABLE1   VBUILD    TYPE=BNCOSMAP
*******************************************************************
* Adjacent network ID                                             *
*******************************************************************
NETWORKA NETWORK   NETID=NETWORKA         adjacent network ID
COS1     MAPSTO    COS=COSA               map COS
COS2     MAPSTO    COS=COSB               map COS
COS3     MAPSTO    COS=COSC               map COS
*******************************************************************
* Adjacent network ID                                             *
*******************************************************************
NETWORKB NETWORK   NETID=(NETWORKB,NETWORKC)  adjacent netw ID
COS1     MAPSTO    COS=COSA               map COS
COS3     MAPSTO    COS=COSF               map COS
COS5     MAPSTO    COS=COSE               map COS
*******************************************************************
* Default network ID                                              *
*******************************************************************
DEFAULT  NETWORK                          default network ID
COS1     MAPSTO    COS=COSA               map COS
```

Figure 4. COS Mapping Table Example

Note: If a request is to be sent to a nonnative network node and the nonnative node does not have the capability to perform mapping, VTAM will map the native COS to a nonnative COS. VTAM will search the native COS names (specified by the COS operand) under the NETWORK definitions for the destination NETID or the default definitions (if the NETID is not defined on a NETWORK statement). Using the first definition of the native COS name, the *cos_name* will be used on the outbound request.

Table 47. Definition Statements for COSMAP Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=BNCOSMAP	529	R			
name	NETWORK	NETID	529				
cos_name	MAPSTO	COS	528				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in a border node COS mapping table.

NETWORK Definition Statement

Code a NETWORK definition statement for each network for which you are defining a border node COS mapping table.

MAPSTO Definition Statement

Code a MAPSTO definition statement to define the native and nonnative APPN COS mappings.

Full Syntax

The full syntax for the border node COS mapping definition follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=BNCOSMAP──────────────────────────────────►◄
    └─name─────┘
```

NETWORK

```
►►──┬──────────┬──NETWORK──┬──────────────────────────────────────────┬──►◄
    └─name─────┘           └─NETID=──┬─network_id──────────┬──────────┘
                                     └─(─network_id──────)─┘
```

MAPSTO

```
►►──cos_name──MAPSTO──COS=native_cosname──────────────────────────────────►◄
```

Operand Descriptions

The following section describes the operands you can code for COS mapping tables.

COS

```
►►──COS=native_cosname────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies the 1–8 character name identifying a native COS name.

cos_name

```
►►──cos_name──────────────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies the 1–8 character name identifying the nonnative COS name.

name

```
►►──┬──────┬──────────────────────────────────────────────────────►◄
    └name──┘
```

statements:	*VBUILD, NETWORK*
dependencies:	*none*

On VBUILD: Assigns a name to border node COS mapping definitions.

On NETWORK: Assigns a name to the NETWORK definition statement. To avoid confusion, use the same value for *name* and the NETID operand.

NETID

```
►►──┬──────────────────────────────┬──────────────────────────────►◄
    └NETID=──┬─network_id─────────┬─┘
             └─(─network_id──────)─┘
```

statements:	*NETWORK*
dependencies:	*none*

Specifies the optional 1–8 character network identifier of the adjacent network. If you do not code NETID at all, this NETWORK definition statement represents a default border node COS mapping to be used for unknown or unlisted NETIDs (those not explicitly coded on other NETWORK statements).

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

TYPE

```
►►──TYPE=BNCOSMAP──────────────────────────────────────────────────►◄
```

statements:	*VBUILD*
dependencies:	*none*

Means that this VBUILD definition statement defines a set of border node COS mapping definitions.

Network Node Server List

You define a network node server list for an end node (EN) with a VBUILD definition statement followed by one or more NETSRVR definition statements.

If you choose not to define a network node server list, a default network node server list consisting of a single nameless NETSRVR definition statement is automatically activated when the end node is initialized.

If the NNSPREF start option is defined, the network node specified will have a higher priority then any other network node specified in the active NETSRVR list.

Table 48. Definition Statement for Network Node Server List

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=NETSRVR	532	R			
		ORDER	531			FIRST	
name	NETSRVR	NETID	531				
		SLUINIT	532			REQ	

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement to begin the network node server list.

NETSRVR Definition Statement

Code a NETSRVR definition statement to define a network node (NN) as a network node server to the EN. There is an unlimited number of NETSRVR definition statements in any list (until VTAM runs out of storage). Duplicate entries in the list are ignored.

If you do not code a NETSRVR definition statement, the end node selects a network node in the same way as when no network node server list is specified.

Full Syntax

The full syntax for the network node server list follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

```
►►──┬──────┬──VBUILD──TYPE=NETSRVR──┬─ORDER=FIRST──────────┬──────────────────────────────►◄
    └name──┘                        │                      │
                                    └ORDER=─┬─FIRST─┬────────┘
                                            └─NEXT──┘
```

```
►►──┬──────┬──NETSRVR──┬──────────────────┬──┬─SLUINIT=REQ──────────┬──────────────────────►◄
    └name──┘           └NETID=network_id──┘  │                      │
                                             └SLUINIT=─┬─REQ─┬────────┘
                                                       └─OPT─┘
```

Operand Descriptions

The following section describes the operands you can code for the network node server list.

name

```
►►──┬──────────┬──────────────────────────────────────────────────►◄
    └─name─────┘
```

On VBUILD: Assigns an optional name to a network node server list.

On NETSRVR: Identifies the CP name of a NN that can be used as a network node server for this EN.

If you do not specify a name on a NETSRVR definition statement, any other known adjacent NN, not included in this network node server list, will be considered as a potential network node server for this EN. Only one nameless NETSRVR definition statement entry may appear in a network node server list, and it must be the last entry in the list. Any subsequent entries in the list are ignored. Note that any named NETSRVR entry previously considered and rejected as a network node server is not tried again during nameless entry processing.

NETID

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─NETID=network_id─┘
```

Specifies a network ID for the named NN. If the NETID operand is not specified, the network ID of this EN is assumed.

Note: NETID is not valid when no value is specified for *name*.

ORDER

```
                ┌─ORDER=FIRST──────┐
►►──────────────┼──────────────────┼──────────────────────────────►◄
                │          ┌─FIRST─┐│
                └─ORDER=───┼───────┼┘
                           └─NEXT──┘
```

Determines how the network node server list is processed.

ORDER=FIRST
Specifies that the EN always attempts to find a network node server from the network node server list starting with the first entry. If the first NN in the list is unavailable, the EN attempts to get the second NN in the list to act as its network node server. If the second NN is also unavailable, the third NN in the list is checked, and so on until an available NN is found. Thus, a prioritized list is created where the most preferred NN server is the first entry in the list, the second most preferred is the second entry in the list, and so on. FIRST is the default order of selection.

ORDER=NEXT
Specifies that the EN attempts to find a network node server starting with the next entry after the NN selected the last time the list was used. If the next NN in the list is unavailable, the EN attempts to get the following NN in the list to act as its network node server, and so on until an available NN is found. When the bottom of the list is encountered, the first entry in the list is considered to be the next entry. Thus, NNs are selected in a round-robin manner and no preference is given to one node in the list over another node.

SLUINIT

```
►►─┬─SLUINIT=REQ──────────┬──────────────────────────────────────────►◄
   │                      │
   └─SLUINIT=─┬─REQ─┬─────┘
              └─OPT─┘
```

Specifies whether the network node server is required to support SLU-initiated sessions as a prerequisite for establishment of CP-CP sessions between the indicated network node and this end node.

SLUINIT=OPT
Specifies that CP-CP sessions between this end node and the indicated network node do not depend on the level of network node server support provided by the network node for SLU-initiated sessions.

SLUINIT=REQ
Specifies that CP-CP sessions between this end node and the indicated network node are not permitted when, during session setup, the network node indicates that its network node server support does not include SLU-initiated session capability.

TYPE

```
►►─TYPE=NETSRVR──────────────────────────────────────────────────────►◄
```

Means that this VBUILD definition statement defines a network node server list.

Chapter 4. Start Options

Start Options

About This Chapter

This chapter describes the VTAM start options and how to code them.

Refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on the following topics:
- Start option sources
- Creating start option and configuration lists
- Overriding start options
- Entering start options during the start procedure.

Coding VTAM Start Options

This section explains the rules used to code start options.

Code start option lists as one or more 80-byte records in the following format:

position 1	**position 72**
[...]item[,item]...	**...c**

... Indicates blanks. One or more blanks can precede the first item.

item Represents a start option in a start option list. Separate items with commas. Use no intervening blanks.

c Reserved for the continuation indicator. The continuation indicator is optional, but if used it can be any nonblank character. VTAM ignores positions 73–80.

Following is a sample start list, ATCSTRyy. The first line is a scale to show you column positions. Every tenth column position is indicated by a number.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
SSCPID=01,
SSCPNAME=SSCP1A
NETID=NETA,
DYNASSCP=YES
SSCPDYN=YES,
HOSTSA=1
```

One start list cannot reference another, and only an operator can enter LIST=yy on the start command to use this start list.

Depending on the number of items, two or more records might be required for the start option list. Observe the following rules if more than one record is required:
- You can code data through position 71, even if you are in the middle of an item. The remainder of the item can be continued in the next record.
- If you do not code data through position 71, end the data in one record with an item, and begin the next record with a new item.
- To create a comment line, enter an asterisk (*) in position 1 of any record.

© Copyright IBM Corp. 1984, 1999

Using MVS System Symbols in VTAM Start Options

You can use the following MVS system symbols in the values you specify for VTAM start options:

&SYSCLONE
Represents a 1–2-character shorthand notation for the name of the system

&SYSNAME
Represents a 1–8-character system name

&SYSPLEX
Represents a 1–8-character sysplex name

&installation-defined system symbol
Represents a 1–8-character identifier. You can define this symbol to be any identifier you want.

The substitution text for these system symbols is defined in MVS and is substituted by MVS in place of the symbols during VTAM startup. Refer to *MVS/ESA SP V5 Initialization and Tuning Reference* for information about how to define the symbols in MVS.

MVS system symbols can be used to allow a single start option list to be used to start multiple VTAMs in a multisystem environment. For information about how to use MVS system symbols in VTAMLST, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Start Options Syntax Diagrams

Options:

```
├──NETID=network_id──SSCPID=sscp_id──SSCPNAME=name──┬─ALSREQ=NO──────────┬──►
                                                    └─ALSREQ=──┬─YES─┬────┘
                                                               └─NO──┘

                    (1)
►──┬─APPNCOS=NONE──────────────────┬──┬─ASIRFMSG=OLUSSCP──────┬──►
   └─APPNCOS=class-of-service_name─┘  └─ASIRFMSG=──┬─ALLSSCP─┬──┘
                                                   ├─OLUSSCP─┤
                                                   └─NONE────┘

   ┌─ASYDE=TERM──────┐  ┌─AUTHLEN=YES──────┐  ┌─AUTORTRY=AUTOCAP──────┐
►──┤                 ├──┤                  ├──┤                       ├──►
   └─ASYDE=──┬─KEEP─┬─┘  └─AUTHLEN=──┬─NO──┬─┘  └─AUTORTRY=──┬─AUTOCAP─┬──┘
             └─TERM─┘                └─YES─┘                 ├─CDRM────┤
                                                             ├─ALL─────┤
                                                             └─NONE────┘
```

```
                                   (2)                    (3)
 ┌─AUTOTI=0──────────┐  ┌─BN=NO──────┐  ┌─BNDYN=LIMITED──────┐
 ├──────────────────┤  ├────────────┤  ├────────────────────┤
 └─AUTOTI=time_period─┘  └─BN=──┬─NO──┬─┘  └─BNDYN=──┬─LIMITED─┬─┘
                               └─YES─┘                ├─NONE────┤
                                                      └─FULL────┘

                    (4)
 ┌─BNORD=PRIORITY─────┐  ┌─BSCMDRS=(STATS,INOPS)──────────────┐
 ├────────────────────┤  ├────────────────────────────────────┤
 └─BNORD=──┬─PRIORITY─┬─┘  └─BSCMDRS=──┬─STATS──────────────────┬─┘
           └─DEFINED──┘                ├─NOSTATS────────────────┤
                                       ├─(,INOPS)───────────────┤
                                       ├─(,NOINOPS)─────────────┤
                                       └─(──┬─STATS──┬─,INOPS──┬──┬─)─┘
                                            │        └─,NOINOPS─┘  │
                                            └─NOSTATS──┬─,INOPS──┬──┘
                                                       └─,NOINOPS─┘

 ┌─BSCTMOUT=286──────────┐
 ├──────────────────────┤  └─poolname=──┬─()──────────────────┬─┘
 └─BSCTMOUT=units_of_time─┘              └─┤ Buffer Pool Values ├─┘

 ┌─CACHETI=8──────────────┐  ┌─CDRDYN=YES──────┐
 ├───────────────────────┤  ├─────────────────┤
 └─CACHETI=number_of_minutes─┘  └─CDRDYN=──┬─NO──┬─┘
                                           └─YES─┘

                                   (5)
 ┌─CDRSCTI=8M──────────┐  ┌─CDSERVR=NO──────┐
 ├────────────────────┤  ├─────────────────┤
 └─CDRSCTI=timeout_value─┘  └─CDSERVR=──┬─YES─┬─┘
                                        └─NO──┘

                 (6)
 ┌─CDSREFER=1──────────────┐  ┌─CINDXSIZ=8176──────┐
 ├────────────────────────┤  ├────────────────────┤
 └─CDSREFER=number_of_CDSs─┘  └─CINDXSIZ=table_size─┘

                 (7)
 ┌─CMPMIPS=100──────────────┐  ┌─CMPVTAM=0──────────────┐
 ├─────────────────────────┤  ├────────────────────────┤
 └─CMPMIPS=compression_ratio─┘  └─CMPVTAM=compression_level─┘

 ┌─CNMTAB=ISTMGC00──────┐  ┌─COLD─┐  ┌─CONFIG=00──────────┐
 ├──────────────────────┤  ├──────┤  ├────────────────────┤
 └─CNMTAB=cnm_table_name─┘  └─WARM─┘  └─CONFIG=──┬─file_id───┬─┘
                                                 └─file_name─┘
```

Start Options

```
                    (8)                                   (9)
 ┌─CONNTYPE=APPN──────┐  ┌─CPCDRSC=NO──────┐  ┌─CPCP=LEASED──────────┐
 │                    │  │                 │  │                      │
 └─CONNTYPE=──┬─APPN─┬─┘  └─CPCDRSC=──┬─YES─┬─┘  └─CPCP=──┬─YES──────┬─┘
              └─LEN──┘                └─NO──┘             ├─NO───────┤
                                                          ├─LEASED───┤
                                                          └─SWITCHED─┘

 ┌─CSALIMIT=0──────┐  ┌─CSA24=0──────┐  ┌─DATEFORM=MDY──────┐
 │                 │  │              │  │                   │
 └─CSALIMIT=value──┘  └─CSA24=value──┘  └─DATEFORM=──┬─MDY─┬─┘
                                                     ├─DMY─┤
                                                     └─YMD─┘

                                   (10)
 ┌─DIALRTRY=YES──────┐  ┌─DIRSIZE=0──────────────────┐
 │                   │  │                            │
 └─DIALRTRY=──┬─NO──┬─┘  └─DIRSIZE=number_of_resources┘
              └─YES─┘

                (11)
 ┌─DIRTIME=8D──────────┐
 │                     │
 └─DIRTIME=time_period─┘

 ┌─DISCNTIM=(15,0)──────────────────┐  ┌─DLRORDER=STATNID──────┐
 │                                  │  │                       │
 └─DISCNTIM=(time_period1,time_period2)┘  └─DLRORDER=──┬─STATNID─┬─┘
                                                       └─CPNAME──┘

                                                   (12)
 ┌─DLRTCB=32──────────────────────────┐  ┌─DSPLYDEF=100──────────────┐
 │                                    │  │                           │
 └─DLRTCB=number_of_task_control_blocks┘  └─DSPLYDEF=number_of_messages┘

 ┌─DSPLYMAX=65535──────────────┐  ┌─DSPLYWLD=FULLWILD──────┐
 │                             │  │                        │
 └─DSPLYMAX=number_of_messages─┘  └─DSPLYWLD=──┬─FULLWILD─┬─┘
                                               ├─NOWILD───┤
                                               ├─OPERONLY─┤
                                               └─POAONLY──┘

                                   (13)
 ┌─DUPDEFS=ALL──────────┐  ┌─DYNADJCP=YES──────┐  ┌─DYNASSCP=YES──────┐
 │                      │  │                   │  │                   │
 └─DUPDEFS=──┬─ALL───┬─┘  └─DYNADJCP=──┬─YES─┬─┘  └─DYNASSCP=──┬─YES─┬─┘
             ├─NONE──┤                └─NO──┘                   └─NO──┘
             ├─APPL──┤
             └─DEPLU─┘
```

```
►─┬─DYNDLGMD=NONE──────────────────┬─┬──────────────────────────────┬──►
  │                                │ │                              │
  └─DYNDLGMD=──┬─logmode_name─┬────┘ │                    (14)      │
               └─NONE─────────┘      └─DYNHPPFX=──prefix────────────┘

►─┬─DYNLU=NO──────────┬─┬─DYNMODTB=NONE──────────────┬─┬─DYNPUPFX=CN──────────────┬──►
  │                   │ │                            │ │                          │
  └─DYNLU=──┬─YES─┬───┘ └─DYNMODTB=──┬─table_name─┬──┘ └─DYNPUPFX=──prefix¹⁴──────┘
            └─NO──┘                  └─NONE───────┘

►──┬──────────────────────────┬─┬─ENCRPREF=NONE──────────┬─┬─ENCRYPTN=YES──────────┬──►
   │                          │ │                        │ │                        │
   └─DYNVNPFX=──prefix¹⁴──────┘ └─ENCRPREF=──┬─NONE─┬─────┘ └─ENCRYPTN=──┬─YES──┬──┘
                                             └─value┘                     ├─NO───┤
                                                                          ├─CCA──┤
                                                                          ├─CUSP─┤
                                                                          ├─24───┤
                                                                          └─31───┘

►─┬─ENHADDR=NO──────────┬──────────────────────────────────────────────────────►
  │                      │
  └─ENHADDR=──┬─NO─┬─────┘
              └─YES┘

►─┬─ESIRFMSG=ALLSSCP──────────┬─┬─FLDTAB=ISTMSFLD──────────┬──────────────────►
  │                           │ │                          │
  └─ESIRFMSG=──┬─ALLSSCP─┬────┘ └─FLDTAB=──┬─table_name─┬──┘
               ├─OLUSSCP─┤                 └─NONE───────┘
               └─NONE────┘

►─┬─FSIRFMSG=OLUSSCP──────────┬─┬─GWSSCP=YES──────────┬─┬─HNTSIZE=4080──────────┬──►
  │                           │ │                     │ │                       │
  └─FSIRFMSG=──┬─ALLSSCP─┬────┘ └─GWSSCP=──┬─YES─┬────┘ └─HNTSIZE=4080──────────┘
               ├─OLUSSCP─┤                 └─NO──┘
               └─NONE────┘

►─┬─HOSTPU=ISTPUS──────────────────┬─┬─HOSTSA=1──────────────────────────┬──────►
  │                                │ │                                   │
  └─HOSTPU=host_subarea_pu_name────┘ │                          (15)     │
                                     └─HOSTSA=subarea_number─────────────┘

                                                                (16)
►─┬─HOTIOTRM=0──────────────────────────┬─┬─HPR=RTP──────────────────────┬──────►
  │                                     │ │                              │
  └─HOTIOTRM=percentage_of_buffer_pool──┘ └─HPR=──┬─ANR──────────┬─────────┘
                                                  ├─RTP──────────┤
                                                  ├─NONE─────────┤
                                                  ├─(RTP,ANR)────┤
                                                  ├─(RTP,NONE)───┤
                                                  └─(ANR,NONE)───┘
```

Start Options

```
►─┬─HPRNCPBF=NO──────────┬──────────────────────────────────────────►
  │                      │
  └─HPRNCPBF=──┬─YES─┬──┘
               └─NO──┘

►─┬─HPRPST=(8M,4M,2M,1M)──────────────────────────────────┬──────────►
  │                                                       │
  └─HPRPST=(low_limit,med_limit,high_limit,network_limit)─┘

                                                   (17)
►─┬─HSRTSIZE=9973──────────────────┬─┬─INITDB=ALL──────┬──────────►
  │                                │ │                 │
  └─HSRTSIZE=number_of_queue_pointers┘ └─INITDB=──┬─ALL──┬─┘
                                                  ├─DIR──┤
                                                  ├─TOPO─┤
                                                  └─NONE─┘

►─┬─INOPDUMP=OFF──────────┬─┬─IOINT=180──────────────┬──────────────►
  │                       │ │                        │
  └─INOPDUMP=──┬─ON──┬──┘ └─IOINT=number_of_seconds──┘
               └─OFF─┘

►─┬─IOMSGLIM=2147483647──────────┬─┬─IOPURGE=0──────────────┬──────►
  │                              │ │                        │
  └─IOMSGLIM=number_of_message_pairs┘ └─IOPURGE=timeout_value┘

►─┬─IPADDR=address─┬──────────────────────────────────────────────►
  │                │
  └────────────────┘

►─┬─IRNSTRGE=0──────┬─┬─ISTCOSDF=INDLU──────────────┬──────────────►
  │                 │ │                             │
  └─IRNSTRGE=size───┘ └─ISTCOSDF=──┬─ALL──────────┬─┘
                                   ├─APPL─────────┤
                                   ├─DEPLU────────┤
                                   ├─INDLU────────┤
                                   ├─NONE─────────┤
                                   └─(──┬─APPL──┬──)─┘
                                        ├─DEPLU─┤
                                        └─INDLU─┘

                                          (18)
►─┬─LIMINTCP=number_of_seconds─┬─┬─LIST=00──────────────────┬──────►
  │                            │ │                          │
  └────────────────────────────┘ └─LIST=start_option_list_id┘
```

```
►──┬──────────────────────────────────────────────────────────────────────────►
   │                                   (19)
   └─LISTBKUP=──┬─start_option_list_id─┬──────────┘
                ├─DEFAULTS─────────────┤
                └─PROMPT───────────────┘

►──┬──────────────────────────────────────────────────────────────────────────►
   └─MAINTLVL=maintenance_level─┘

                 (20)
   ┌─MAXLOCAT=5000──────────────────────────┐   ┌─MAXLURU=6144──────┐
►──┤                                        ├───┤                   ├──────────►
   └─MAXLOCAT=max_locate_congestion_threshhold─┘   └─MAXLURU=ru_length─┘

   ┌─MAXSSCPS=10──────────────┐   ┌─MAXSUBA=15──────────────────┐
►──┤                          ├───┤                             ├──────────────►
   └─MAXSSCPS=number_of_SSCPs─┘   └─MAXSUBA=maximum_subarea_number─┘

   ┌─MIHTMOUT=1800──────────┐   ┌─MSGLEVEL=BASE──────┐   ┌─MSGMOD=NO──────┐
►──┤                        ├───┤                    ├───┤                ├──────►
   └─MIHTMOUT=units_of_time─┘   └─MSGLEVEL=──┬─BASE─┬─┘   └─MSGMOD=──┬─NO─┬─┘
                                             ├─V4R1─┤                  └─YES┘
                                             ├─V4R2─┤
                                             ├─V4R3─┤
                                             ├─V4R4─┤
                                             └─CS390┘

   ┌─MXSAWBUF=10000──────────┐   ┌─MXSSCPRU=4096──────┐
►──┤                         ├───┤                    ├──────────────────────┤
   └─MXSAWBUF=number_of_buffers─┘   └─MXSSCPRU=ru_length─┘
```

Notes:

1 APPNCOS is meaningful only if the NODETYPE start option is also used.

2 BN is meaningful only if the NODETYPE=NN start option is also used.

3 BNDYN is meaningful only if the BN=YES start option is also used.

4 BNORD is meaningful only if the BN=YES start option is also used.

5 CDSERVR is meaningful only if the NODETYPE=NN start option is also used.

6 CDSREFER is meaningful only if the NODETYPE=NN and CDSERVR=NO start options are also used.

7 The CMPMIPS start option is meaningful only if the value for CMPVTAM is greater than 1.

8 CONNTYPE is meaningful only if the NODETYPE start option is also used.

9 CPCP is meaningful only if the NODETYPE start option is also used.

10 DIRSIZE is meaningful only if the NODETYPE=NN start option is also used.

11 DIRTIME is meaningful only if the NODETYPE=NN start option is also used.

12 If the DSPLYMAX start option value is less than 100, that value is the default for DSPLYDEF.

13 DYNADJCP is meaningful only if the NODETYPE start option is also used.

14 Two character prefix.

15 HOSTSA specifies the subarea number of this VTAM. If HOSTSA is not coded, then a default subarea number of 1 is used.

16 HPR is meaningful only if NODETYPE is also used.

17 INITDB is meaningful only if the NODETYPE=NN start option is also used.

18 LIST can be entered by a VTAM operator only. If LIST is coded in an ATCSTR *-xx* file, it is considered to be an error and is ignored.

19 LISTBKUP can only be coded in a start option file. If you enter it on the START command or at an operator prompt, VTAM will ignore it.

20 MAXLOCAT is meaningful only if NODETYPE is specified.

```
|-┌MXSUBNUM=511──────────┐──NETID=network_id──┌NCPBUFSZ=512──────────┐──────────►
  └MXSUBNUM=──┬─255──┬──┘                    └NCPBUFSZ=──┬─512──┬──┘
              ├─511──┤                                    ├─1024─┤
              ├─1023─┤                                    └─2048─┘
              ├─2047─┤
              ├─4095─┤
              ├─8191─┤
              ├─16383┤
              ├─32767┤
              └─65535┘

                                                (1)
►──┌NMVTLOG=NPDA──────────┐──┌NNSPREF=NONE──────────────────┐──┬──────────────┬──►
   └NMVTLOG=──┬─ALWAYS─┬──┘  └NNSPREF=network_node_server──┘  └NODELST=name───┘
              ├─NEVER──┤
              └─NPDA───┘

►──┬──────────────────────────┬──┌NQNMODE=NAME──────────┐──────────────────────►
   │                    (2)   │  └NQNMODE=──┬─NAME───┬──┘
   └NODETYPE=──┬─EN─┬──────────┘             └─NQNAME─┘
               └─NN─┘

►──┌NUMTREES=100──────────────────────────┐──┬──────────────────────────┬──────►
   │                                  (3) │  └NSRTSIZE=(─netid,size──)─┘
   └NUMTREES=number_of_routing_trees──────┘

►──┌OSIEVENT=PATTERNS──────┐──┌OSIMGMT=NO──────────┐──┌OSITOPO=(ILUCDRSC)──────┐──►
   └OSIEVENT=──┬─ALL─────┬──┘  └OSIMGMT=──┬─YES─┬──┘  │                        │
               ├─NONE────┤                └─NO──┘     └OSITOPO=──┬─ALLCDRSC─┬──┘
               └─PATTERNS┘                               └─ILUCDRSC─┘
```

```
►─┬─OSRTSIZE=43──────────────────────────┬─┬─PDTRCBUF=2──────────────────┬──►
  │                                      │ │                             │
  └─OSRTSIZE=number_of_queue_pointers────┘ └─PDTRCBUF=number_of_buffers────┘

  ┌─PIUMAXDS=200──────────────┐  ┌─PLUALMSG=NOSUPP──────────┐
►─┤                           ├──┤                          ├──────────────►
  └─PIUMAXDS=calculation_factor┘  └─PLUALMSG=──┬─SUPPRESS─┬─┘
                                               └─NOSUPP───┘

  ┌─PPOLOG=NO──────────┐
►─┤                    ├──────────────────────────────────────────────────►
  └─PPOLOG=──┬─NO──┬──┘
             └─YES─┘

            (4)
  ┌─PROMPT──────┐
►─┤             ├──────────────────────────────────────────────────────────►
  └─NOPROMPT────┘

  ┌─PSRETRY=(0,0,0,0)──────────────────────────────────────────┐
►─┤                                                            ├──────────►
  └─PSRETRY=(low_tp_int,med_tp_int,high_tp_int,network_tp_int)─┘

                                                                   (5)
  ┌─PSSTRACE=NORB──────┐  ┌─PSWEIGHT=LESSTHAN──────┐  ┌─RESUSAGE=100──────────┐
►─┤                    ├──┤                        ├──┤                       ├──►
  └─PSSTRACE=──┬─NORB─┬─┘  └─PSWEIGHT=──┬─EQUAL────┬─┘  └─RESUSAGE=usage_limit─┘
               ├─IRB──┤                ├─LESSTHAN─┤
               ├─SRB──┤                └─SAMEROUT─┘
               └─BOTH─┘

                (6)
  ┌─ROUTERES=128──────────────┐
►─┤                           ├──────────────────────────────────────────┤
  └─ROUTERES=resistance_value─┘
```

Notes:

1 NNSPREF can be specified only if NODETYPE=EN is specified during VTAM START processing.

2 NODETYPE enables APPN function. The combination of HOSTSA, NODETYPE, and SACONNS determines the configuration (subarea node, interchange node, migration data host, network node, or end node).

3 NUMTREES is meaningful only if the NODETYPE=NN start option is also used.

4 A VTAM operator cannot enter the PROMPT or NOPROMPT start option; it can be coded only in ATCSTR00. The value coded in ATCSTR00 is ignored if start options are entered on the START command or if VTAM finds an error in a start list. Upon finding an error in a start list, VTAM prompts the operator so that the operator can specify the option correctly.

5 RESUSAGE is meaningful only if the NODETYPE=NN start option is also used.

6 ROUTERES is meaningful only if the NODETYPE=NN start option is also used.

```
                                                  (1)
  ┌─SACONNS=YES──────┐   ┌─SAVERSCV=(NO,KEEP)──────────┐
  │                  │   │                             │
  └─SACONNS=─┬─NO──┬─┘   └─SAVERSCV=─┬─(,FREE)─────────┤
             └─YES─┘                 ├─(,KEEP)─────────┤
                                     ├─NO──────────────┤
                                     ├─(NO,FREE)───────┤
                                     ├─(NO,KEEP)───────┤
                                     ├─YES─────────────┤
                                     ├─(YES,FREE)──────┤
                                     ├─(YES,KEEP)──────┤
                                     ├─SAWDATA─────────┤
                                     ├─(SAWDATA,FREE)──┤
                                     └─(SAWDATA,KEEP)──┘

  ┌─SAWMAXDS=100──────────────┐   ┌─SAWMXQPK=0──────────────┐
  │                           │   │                         │
  └─SAWMAXDS=calculation_factor┘   └─SAWMXQPK=calculation_factor┘

  ┌─SDLCMDRS=(STATS,INOPS)──────────────┐                  (2)
  ├─SDLCMDRS=YES────────────────────────┤   ┌─SECLVLCP=ADAPT──┐
  │                                     │   │                 │
  └─SDLCMDRS=─┬─YES──────────────────┬──┘   └─SECLVLCP=─┬─LEVEL1─┬─┘
              ├─NO───────────────────┤                  ├─LEVEL2─┤
              ├─STATS────────────────┤                  └─ADAPT──┘
              ├─NOSTATS──────────────┤
              ├─(,INOPS)─────────────┤
              ├─(,NOINOPS)───────────┤
              └─(─┬─STATS─┬─,INOPS──┬──┬─)─┘
                  │       └─,NOINOPS┘  │
                  └─NOSTATS─┬─,INOPS──┬┘
                            └─,NOINOPS┘

  ┌─SIRFMSG=ALLSSCP──────┐   ┌─SLOWVAL=(0,0)──────────────┐
  │                      │   │                            │
  └─SIRFMSG=─┬─ALLSSCP─┬─┘   └─SLOWVAL=─┬─warn_time──────┬─┘
             ├─OLUSSCP─┤                └─(warn_time,end_time)┘
             └─NONE────┘

  ┌─SLUALMSG=NOSUPP──────┐   ┌─SMEAUTH=DISCARD──────┐
  │                      │   │                      │
  └─SLUALMSG=─┬─SUPPRESS─┬─┘   └─SMEAUTH=─┬─DISCARD─┬─┘
              └─NOSUPP───┘                ├─EXIT────┤
                                          └─KEEP────┘
```

```
                                                  (3)
 ┌─SNAPREQ=1000──────────────┐  ┌─SNVC=3──────────────────┐
►┤                           ├──┤                         ├──────────────────►
 └─SNAPREQ=number_of_requests┘  └─SNVC=subnet_visit_count┘

                                                                  (4)
 ┌─SONLIM=(60,30)──────────────────┐  ┌─SORDER=APPN──────────┐
►┤                                 ├──┤                      ├──────────────►
 └─SONLIM=──┬─(percent_1,percent_2)┬┘  └─SORDER=──┬─ADJSSCP──┬┘
            ├─(percent_1)──────────┤               ├─APPN──────┤
            └─(,percent_2)─────────┘               ├─APPNFRST──┤
                                                   └─SUBAREA───┘

                                                  (5)
 ┌─SRCHRED=OFF──────┐  ┌─SRCOUNT=10──────────────────────┐
►┤                  ├──┤                                 ├──────────────────►
 └─SRCHRED=──┬─OFF──┬┘  └─SRCOUNT=number_of_search_requests┘
             └─ON───┘

                (6)                          (7)
 ┌─SRTIMER=30──────────────┐  ┌─SSCPDYN=YES──────┐
►┤                         ├──┤                  ├──────────────────────────►
 └─SRTIMER=number_of_seconds┘  └─SSCPDYN=──┬─NO───┬┘
                                           └─YES──┘

 ┌─SSCPORD=PRIORITY──────────┐  ┌─SSDTMOUT=30──────────────┐
►┤                           ├──┤                          ├──────────────────►
 └─SSCPORD=──┬─DEFINED───────┬┘  └─SSDTMOUT=number_of_seconds┘
             └─PRIORITY──────┘

                 (8)
 ┌─SSEARCH=YES──────────────┐  ┌─STRGR=ISTGENERIC──────────┐
►┤                          ├──┤                           ├──────────────────►
 └─SSEARCH=──┬─YES──────────┬┘  └─STRGR=──┬─ISTGENERIC────┬┘
             ├─NO───────────┤             ├─NONE──────────┤
             ├─CACHE────────┤             └─structure_name─┘
             └─APPNFRST─────┘

 ┌─STRMNPS=ISTMNPS──────────────┐  ┌─SUPP=NOSUP──────────┐
►┤                              ├──┤                     ├──────────────────►
 └─STRMNPS=──┬─ISTMNPS─────────┬┘  └─SUPP=──┬─NOSUP──┬┘
             ├─NONE───────────┤             ├─INFO───┤
             └─structure_name─┘             ├─WARN───┤
                                            ├─NORM───┤
                                            └─SER────┘

 ┌─SWNORDER=CPNAME──────┐
►┤                      ├──────────────────────────────────────────────────┤
 └─SWNORDER=──┬─CPNAME──┬┘
              └─STATNID─┘
```

Start Options

Notes:

1 SAVERSCV is meaningful only if NODETYPE is also used.

2 The SECLVLCP start option is meaningful only if the NODETYPE and VERIFYCP start options are also used.

3 SNVC is meaningful only if the BN=YES start option is also used.

4 SORDER is meaningful only in an interchange node or a migration data host.

5 SRCOUNT is meaningful only if the SRCHRED=ON start option is also used.

6 SRTIMER is meaningful only if the SRCHRED=ON start option is also used.

7 The SSCPDYN start option applies only for interconnected networks (that is, GWSSCP=YES is used).

8 SSEARCH is meaningful only if the NODETYPE=NN start option is also used.

```
├─┬─NOTNSTAT──────────────────────────────┬──────────────────────────────►
  │                                       │
  └─TNSTAT─┬─,NOCNSL─┬──┬─,TIME=60──────┬─┘
           └─,CNSL───┘  └─,TIME=minutes─┘

►─┬─NOTRACE,TYPE=BUF (1)──────────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=BUF─┤ BUF Trace Operands (2) ├─┘

►─┬─NOTRACE,TYPE=IO (1)───────────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=IO─┤ IO Trace Operands (2) ├─┘

►─┬─NOTRACE,TYPE=LINE (1)─────────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=LINE─┤ LINE Trace Operands (2) ├─┘

►─┬─NOTRACE,TYPE=MODULE (1)───────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=MODULE,─┤ MODULE Operands (2) ├─┘

►─┬─NOTRACE,TYPE=SIT (1)──────────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=SIT─┤ SIT Trace Operands (2) ├─┘

►─┬─NOTRACE,TYPE=SMS (1)──────────┬──────────────────────────────────────►
  │                               │
  └─TRACE,TYPE=SMS─,ID=VTAMBUF (2)─┘

►─┬─NOTRACE,TYPE=STATE (1)────────────────┬──────────────────────────────►
  │                                       │
  └─TRACE,TYPE=STATE─┤ STATE Trace Operands (2) ├─┘
```

```
►─┬─TRACE,TYPE=VTAM,─┤ VIT Operands (2) ├─┬──────────────────────────────────┤
  │                                      │
  └─NOTRACE,TYPE=VTAM (3)────────────────┘
```

Notes:

1 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

2 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

3 NOTRACE,TYPE=VTAM is accepted but ignored. Tracing is started with the default trace table size and the default options.

```
├─┬─TRANSLAT=(0,1,2,3,4,5,6,7)──────────┬──────────────────────────────────►
  │                                     │
  └─TRANSLAT=─┬─value──────────────────┬─┘  └─TCPNAME=tcp_job-name─┘
              ├─()─────────────────────┤
              └─(─┬─┬─OLUALIAS─┬─┬─)─┘
                  │ └─0────────┘ │
                  ├─┬─DLUREAL──┬─┤
                  │ └─1────────┘ │
                  ├─┬─DLUSSCP──┬─┤
                  │ └─2────────┘ │
                  ├─┬─COSNAME──┬─┤
                  │ └─3────────┘ │
                  ├─┬─LOGMODES─┬─┤
                  │ └─4────────┘ │
                  ├─┬─LUANAME──┬─┤
                  │ └─5────────┘ │
                  ├─┬─USERVAR──┬─┤
                  │ └─6────────┘ │
                  └─┬─DLUALIAS─┬─┘
                    └─7────────┘
```

```
                (1)                                   (2)
►─┬─UPDDELAY=60──────┬─┬──────────────────────┬─┬─VERIFYCP=NONE──────────┬─►
  │                  │ │                      │ │                        │
  └─UPDDELAY=max_time┘ └─USSTAB=table_name────┘ └─VERIFYCP=─┬─NONE────────┬─┘
                                                            ├─OPTIONAL────┤
                                                            └─REQUIRED────┘

                                        (3)                          (4)
►─┬─VFYRED=YES──────┬─┬─VFYREDTI=OFF──────────────────┬─┬─VRTG=NO──────┬─►
  │                 │ │                               │ │              │
  └─VFYRED=─┬─YES─┬─┘ └─VFYREDTI=─┬─OFF──────────────┬─┘ └─VRTG=─┬─YES─┬─┘
            └─NO──┘               ├─0────────────────┤           └─NO──┘
                                  └─reduction_timer──┘
```

Start Options

```
                    (5)
►─┬─VRTGCPCP=YES──────────┬─┬─VTAMEAS=32001──────────────────────────┬──►
  │                       │ │                                        │
  └─VRTGCPCP=──┬─YES─┬────┘ └─VTAMEAS=number_of_concurrent_sessions──┘
               └─NO──┘

►─┬─XCFINIT=YES───────────┬─┬─XNETALS=NO──────────┬──────────────────────┤
  │                       │ │                     │
  └─XCFINIT=──┬─NO──┬─────┘ └─XNETALS=──┬─NO──┬───┘
              └─YES─┘                   └─YES─┘
```

Buffer Pool Values:

```
├──(──┬──────┬──,──┬──────┬──,──┬──────┬──,──┬───┬──,──┬──────┬──,──►
      └baseno┘     └bufsize┘     └slowpt┘     └F──┘     └xpanno┘

►──┬──────┬──,──┬──────┬──)──────────────────────────────────────────┤
   └xpanpt┘     └xpanlim┘
```

BUF Trace Operands:

```
├──,ID=node_name──┬──────────┬──┬─,AMOUNT=PARTIAL──────────┬──────────►
                  └─,EVERY───┘  │                          │
                                └─,AMOUNT=──┬─FULL────┬────┘
                                            └─PARTIAL─┘

►─┬─,IDTYPE=RESOURCE──────────┬─┬─,SAVE=YES──────────┬──────────────────┤
  │                           │ │                    │
  └─,IDTYPE=──┬─CP────────┬───┘ └─,SAVE=──┬─NO──┬────┘
              ├─SSCP──────┤               └─YES─┘
              └─RESOURCE──┘
```

IO Trace Operands:

```
├──,ID=node_name──┬──────────┬──┬─,IDTYPE=RESOURCE──────────┬─┬─,SAVE=YES──────────┬──┤
                  └─,EVERY───┘  │                           │ │                    │
                                └─,IDTYPE=──┬─CP────────┬───┘ └─,SAVE=──┬─NO──┬────┘
                                            ├─SSCP──────┤               └─YES─┘
                                            └─RESOURCE──┘
```

LINE Trace Operands:

```
├──,ID=line_name──┬─,COUNT=ALL (6)──────────────┬──────────────────────┤
                  │                             │
                  └─,COUNT=──┬─ALL──────────────┬┘
                             └─number_of_bytes──┘
```

MODULE Operands:

```
├──,OPTION=──┬─ALL──────────────┬──────────────────────────────────────┤
             ├─option───────────┤
             ├─(option)─────────┤
             └─(──┬─COMMAND────────┬─)─┘
                  ├─CONNECTIONS────┤
                  ├─DEFINITION─────┤
                  ├─INTERFACES─────┤
                  ├─MANAGEMENT─────┤
                  ├─NOEXIT─────────┤
                  └─SESSION────────┘
```

SIT Trace Operands:

```
                      ┌─,COUNT=ALL (6)──────────┐
├──,ID=line_name──────┤                         ├──┬──────────────────────────┬──┤
                      └─,COUNT=──┬─ALL──────────┘  └─,TRACEPT=trace_point_id─┘
                                 └─number_of_bytes─┘
```

STATE Trace Operands:

```
                      ┌─,IDTYPE=RESOURCE──────┐
├──┬─,ID=node_name────┤                       ├──┬──────────────────────────┤
   │                  └─,IDTYPE=──┬─CP────────┤  │
   │                              ├─SSCP──────┤  │
   │                              └─RESOURCE──┘  │
   └─,OPTION=──┬─ALL──────────────┬──────────────┘
               ├─option───────────┤
               ├─(option)─────────┤
               └─(──┬─ADJCP────────┬─)─┘
                    ├─APPL─────────┤
                    ├─CDRM─────────┤
                    ├─CDRSC────────┤
                    ├─GROUP────────┤
                    ├─LINE─────────┤
                    ├─LNKS─────────┤
                    ├─LU───────────┤
                    ├─NCP──────────┤
                    └─PU───────────┘
```

VIT Operands:

```
   ┌─,MODE=INT,SIZE=100,DSPSIZE=0──────────────────────┐
├──┤                                                   ├──────────────────────►
   │                   ┌─,BFRNUM=2──────┐              │
   └─,MODE=──┬─EXT──────┤                ├──────────────┤
             │          └─,BFRNUM=number─┘              │
             └─INT──┬──────────────────────┬──┬──────────────────────┬──┘
                    └─,SIZE=size───────────┘  └─,DSPSIZE=dspsize─────┘
```

Start Options

```
                                                  (7)  (8)
  ┌─,OPTION=(API,CIO,MSG,NRM,PIU,PSS,SMS,SSCP)──────────┐
►─┼──────────────────────────────────────────────────────┼──────────────────────┤
  ├─,SUBTRACE=ARBP,OPTION=─┬─HPR──────────────────────────┤
  │                        ├─ALL──────────────────────────┤
  │                        │                 (9)          │
  │                        └─(..,HPR,..)────────────────┘
  ├─,SUBTRACE=TREE,OPTION=─┬─SSCP─────────────────────────┤
  │                        ├─ALL──────────────────────────┤
  │                        │                 (10)         │
  │                        └─(..,SSCP,..)───────────────┘
  └─,OPTION=─┬─ALL──────────────────────────┐
             ├─NONE─────────────────────────┤
             ├─option───────────────────────┤
             ├─(option)─────────────────────┤
             │      ┌─,──────┐             │
             └─(────┴─┬─API──┬─────────)──┘
                      ├─APPC─┤
                      ├─CFS──┤
                      ├─CIA──┤
                      ├─CIO──┤
                      ├─CMIP─┤
                      ├─CSM──┤
                      ├─ESC──┤
                      ├─HPR──┤
                      ├─LCS──┤
                      ├─LOCK─┤
                      ├─MSG──┤
                      ├─NRM──┤
                      ├─PIU──┤
                      ├─PSS──┤
                      ├─SMS──┤
                      ├─SSCP─┤
                      ├─TCP──┤
                      ├─VCNS─┤
                      ├─XBUF─┤
                      └─XCF──┘
```

Notes:

1 UPDDELAY is meaningful only if the OSIMGMT=YES start option is also used.

2 The VERIFYCP start option is meaningful only if the NODETYPE start option is also used.

3 VFYREDTI is meaningful only if the NODETYPE=NN start option is also used.

4 VRTG is meaningful only if the NODETYPE and HOSTSA start options are also used.

5 VRTGCPCP is meaningful only if the NODETYPE and HOSTSA start options are also used.

6 COUNT applies only to the IBM 3720 and 3745 Communication Controllers.

7 The default options apply only to MODE=INT.

8 PSS and SMS can be turned off.

9 If multiple trace options are coded in parentheses, HPR must be one of the options coded inside the parentheses when SUBTRACE=ARBP is coded.

10 If multiple trace options are coded in parentheses, SSCP must be one of the options coded inside the parentheses when SUBTRACE=TREE is coded.

Summary of Start Options

The tables in this section show the start options organized by function. Each table includes a brief description of the start option.

The tables show the start options that define or affect:

- SSCP or CP characteristics (Table 49)
- VTAM initialization (Table 50 on page 550)
- Storage (Table 51 on page 551)
- Session control (Table 52 on page 551)
- Connectivity (Table 53 on page 552)
- Session security (Table 54 on page 553)
- Messages (Table 55 on page 553)
- Recording and statistics (Table 56 on page 554)
- Traces and dumps (Table 57 on page 554)
- Buffer pools (Table 58 on page 555)
- Performance and tuning (Table 59 on page 555)
- APPN characteristics (Table 60 on page 557)
- Functions from zappable constants (Table 61 on page 559)

See “Descriptions of Start Options” on page 560 for more detailed descriptions of the start options.

SSCP or CP Characteristics Start Options

Table 49. Start Options That Define SSCP or CP Characteristics. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=SSCPCP command.

Start Option	Description
BN	Whether this node is to provide extended border node function
CDSERVR	Whether this node is a central directory server
DATEFORM	The format that VTAM uses for dates
ENHADDR	Whether high-order element addressing for PLUs can be used.
GWSSCP	Whether VTAM is a gateway SSCP
HOSTPU	The name of the host subarea PU
HOSTSA	The host subarea number
LIMINTCP	The interval to retain a free SNASVCMG session on limited resources for network management flows
MAINTLVL	The maintenance level of the host as defined by the user
MAXSSCPS	The maximum number of SSCPs that are searched on a specific path for the destination SSCP before alternate paths are tried
MAXSUBA	The highest subarea number that can be assigned to any node in this network

Table 49. Start Options That Define SSCP or CP Characteristics (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=SSCPCP command.

Start Option	Description
MXSUBNUM	The maximum subarea number supported by a network to which a VTAM host is interconnected
NETID	The name of the network containing the host
NODETYPE	The APPN node type (network node or end node)
NQNMODE	Whether VTAM defines cross-network resources by their non-network-qualified names in addition to their network-qualified names
SSCPID	The SSCP identifier used when a PU or external CDRM establishes contact with VTAM
SSCPNAME	The name of the VTAM SSCP or CP
STRGR	The name of the generic resources structure
STRMNPS	The name of the structure for multinode persistent sessions
TRANSLAT	Which names are to be translated by alias name translation
USSTAB	The name of the USS table for operator commands and messages

VTAM Initialization Start Options

Table 50. Start Options That Affect VTAM Initialization. These start options (excluding LISTBKUP, NOPROMPT, and PROMPT) can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=VTAMINIT command.

Start Option	Description
COLD	On starting VTAM, restore major nodes to their initial status as defined by the user
CONFIG	The list of major nodes to activate when VTAM is started
INITDB	Whether to load the APPN directory database, topology database, neither, or both when VTAM is started
LIST	The ID of the start option list to use for starting VTAM
LISTBKUP	The action that VTAM should take if an error is found in a start option list
NNSPREF	Specifies the name of the preferred network node server for this end node.
NODELST	The name of the configuration restart file
NOPROMPT	Do not prompt the operator to enter start options
OSIMGMT	Indicates support of CMIP application programs on VTAM, including the VTAM topology agent
PROMPT	Prompt the operator to enter start options
WARM	On starting VTAM, restore major nodes to their status before VTAM's last termination, as recorded in the configuration restart file

Storage Usage Start Options

Table 51. Start Options That Define Storage Usage. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=STORAGE command.

Start Option	Description
CSALIMIT	The maximum amount of common service area (CSA) that can be used by VTAM
CSA24	The maximum amount of 24-bit addressable CSA that can be used by VTAM
DLRTCB	The maximum number of task control blocks in VTAM private storage for dump-load-restart subtasks and file I/O services.
IRNSTRGE	The maximum amount of VTAM private storage that can be used for host intermediate routing node (IRN) transmissions

Session Control Start Options

Table 52. Start Options That Affect Session Control. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=SESSCONT command.

Start Option	Description
APPNCOS	The APPN class of service that is used if the requested class of service cannot be found
ASYDE	Whether to terminate a session when an asynchronous device end RU is received from a local non-SNA device
AUTORTRY	Specifies which adjacent node activation will cause a retry of pending automatic logon requests
BNDYN	Indicates how nodes are added dynamically to adjacent cluster routing lists
BNORD	Indicates the order VTAM does cross-subnet searches
CDRDYN	Specifies whether the host is authorized to dynamically define CDRSC representations of cross-domain resources
CMPMIPS	Use to determine the amount of time for adaptive mode compression
CMPVTAM	The data compression level to be used for sessions with the application programs running on the host
CPCDRSC	Whether VTAM allows applications or LUs to initiate and establish sessions to a dynamic LEN CP ILU
DLRORDER	The way VTAM locates a DLUR PU
DYNADJCP	Whether dynamic entries are created in ISTADJCP for adjacent APPN nodes
DYNASSCP	Whether VTAM dynamically routes session establishment requests to all active adjacent SSCPs if no appropriate adjacent SSCP table is defined
DYNDLGMD	Specifies the default logon mode name used for dynamic cross-domain resources when the resource is acting as the secondary logical unit (SLU) and a logon mode name is not otherwise provided
DYNLU	Whether independent LUs can be dynamically defined
DYNMODTB	Specifies the name of a logon mode table used for dynamic cross-domain resources to correlate each logon mode name with a set of session parameters when the resource acts as the SLU

Table 52. Start Options That Affect Session Control (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=SESSCONT command.

Start Option	Description
HOTIOTRM	The percentage of the I/O buffer pool that can be used by one session before the amount is considered to be excessive and the session is terminated
ISTCOSDF	Indicates the resource types that can use the ISTCOSDF logmode entry.
RESUSAGE	The number of times an APPN resource (node or TG) can be used for routing before it is considered overused
ROUTERES	The relative amount of resistance to having this APPN node perform intermediate session routing
SMEAUTH	Controls session management exit (SME) authorization data without having to modify or code an SME.
SORDER	The order in which to search APPN and subarea networks when a network search request is received from a subarea network
SSCPDYN	Whether VTAM can dynamically add entries to an adjacent SSCP table
SSCPORD	The order in which VTAM searches an adjacent SSCP table
SSEARCH	Whether to search a subarea network when a network search request is received from an APPN network
SWNORDER	The way VTAM locates a switched PU

Connectivity Start Options

Table 53. Start Options That Affect Connectivity. These start options affect session establishment, termination, or search order, and can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=CONNECT command.

Start Option	Description
ALSREQ	Whether adjacent link station names are required to be in the default ALS list of independent LUs
AUTHLEN	Whether VTAM is enabled to pass transmission priority from one LEN node to another
CONNTYPE	Whether a LEN connection or an APPN connection is established with type 2.1 PUs
CPCP	The types of connections on which CP-CP sessions are allowed with adjacent nodes
DISCNTIM	Amount of time VTAM delays deactivation of SSCP-PU session when there are no outstanding LU-LU session requests
HPR	Specifies the level of high performance routing (HPR) support provided by VTAM
HPRNCPBF	Specifies whether HPR can be used although it may cause session data to travel through an NCP twice.
NNSPREF	Specifies the name of the preferred network node server for this end node.
SNVC	Maximum number of networks searched for a resource by this border node
SSDTMOUT	The amount of time that a switched PU can go without a session before the connection is dropped

Table 53. Start Options That Affect Connectivity (continued). These start options affect session establishment, termination, or search order, and can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=CONNECT command.

Start Option	Description
VRTG	Whether virtual-route-based transmission groups connections are requested when SSCP-SSCP session is established
VRTGCPCP	Whether CP-CP sessions are supported over the virtual-route-based transmission group
XCFINIT	Specifies whether XCF connections to other VTAMs in a sysplex are to be established at VTAM initialization.
XNETALS	Whether VTAM can establish sessions with LUs in adjacent networks

Session Security Start Options

Table 54. Start Options That Affect Session Security. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=SECURITY command.

Start Option	Description
ENCRPREF	The prefix to be used for common cryptographic architecture (CCA) key-encrypting key (KEK) labels.
ENCRYPTN	VTAM's level of cryptography support
SECLVLCP	Whether VTAM uses basic or enhanced protocol for session-level security verification during activation of LU 6.2 sessions involving APPN control points
VERIFYCP	Whether VTAM performs LU-LU session-level verification during activation of LU 6.2 sessions involving APPN control points

Messages Start Options

Table 55. Start Options That Affect Messages. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=MESSAGES command.

Start Option	Description
ASIRFMSG	The SSCPs issue messages IST890I and IST896I when an autologon session initiation request fails
CNMTAB	Specifies the name of the communication network management (CNM) routing table used to route unsolicated request units.
DSPLYDEF	Limits the number of messages displayed when certain DISPLAY commands are issued without the MAX operand
DSPLYMAX	Limits the value that can be specified for the DSPLYDEF start option and for the MAX operand on certain DISPLAY commands
DSPLYWLD	Whether wildcards are permitted in certain DISPLAY commands
ESIRFMSG	The SSCPs issue messages IST891I, IST892I, and IST893I when a session initiation request fails and extended sense data exists.
FLDTAB	Whether VTAM uses a message-flooding prevention table to suppress duplicate messages

Table 55. Start Options That Affect Messages (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=MESSAGES command.

Start Option	Description
FSIRFMSG	The SSCPs issue messages IST894I, IST895I, IST1704I, and IST1705I when a session initiation request fails because VTAM is unable to locate the destination LU by using an adjacent SSCP table.
IOMSGLIM	The maximum number of IST530I and IST532I message pairs that are written for individual subareas by the IOPD facility
MSGLEVEL	Whether VTAM is to issue the level of messages containing network-qualified names
MSGMOD	Whether VTAM is to insert in messages the name of the module that issued the message
PLUALMSG	Whether to suppress duplicate session setup failure messages associated with LOGAPPL failures to the same PLU
SIRFMSG	The SSCPs issue IST663I, IST664I, IST889I, and subordinate messages when a session initiation request fails.
SLUALMSG	Whether to suppress messages for session setup failures that occur because the SLU is not enabled
SUPP	The message suppression level for the operator console

Recording and Statistics Start Options

Table 56. Start Options That Affect Recording and Statistics. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=RECSTATS command.

Start Option	Description
BSCMDRS	Which BSC miscellaneous data records are to be recorded in LOGREC
NMVTLOG	Whether to record NMVT alerts in LOGREC
PPOLOG	Whether to send operator commands and their resulting messages to the primary program operator for logging
SDLCMDRS	Which SDLC miscellaneous data records are to be recorded in LOGREC
TNSTAT	Whether, where, and how frequently to record tuning statistics

Traces and Dumps Start Options

Table 57. Start Options That Affect Traces and Dumps. These start options (excluding TRACE) can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=TRACDUMP command.

Start Option	Description
INOPDUMP	Whether to dump VTAM for certain INOP conditions
PSSTRACE	Whether to record IRB and SRB entries in the VTAM internal trace table when the PSS trace option is in effect
SNAPREQ	The number of requests for VTAM buffers between snapshot dumps
TRACE	Which traces to start at VTAM initialization

Buffer Pools Start Options

Table 58. Start Options That Define Buffer Pools. These start options can be displayed as a group with the DISPLAY BFRUSE command.

Start Option	Description
APBUF	The buffer pool that provides fixed common storage with 24-bit addressable buffers
BSBUF	The buffer pool that provides boundary session control block buffers
CRA4BUF	The buffer pool that provides copied request at 4080 addressable buffer size
CRA8BUF	The buffer pool that provides copied request at 8176 addressable buffer size
CRPLBUF	The buffer pool that provides copied request parameter list buffers
IOBUF	The buffer pool that provides input/output buffers
LFBUF	The buffer pool that provides large fixed buffers
LPBUF	The buffer pool that provides large pageable buffers
SFBUF	The buffer pool that provides small fixed buffers
SPBUF	The buffer pool that provides small pageable buffers
TIBUF	The buffer pool that contains control information to support HPDT services.
XDBUF	The buffer pool that provides XID buffers

Performance and Tuning Start Options

Table 59. Start Options That Affect Performance and Tuning. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=PERFTUNE command.

Start Option	Description
AUTOTI	Specifies how often pending automatic logon requests owned by this host are retried
BSCTMOUT	The interval in which two general poll failures are allowed for BSC 3270 terminals
CACHETI	The amount of time that an APPN search request is remembered to prevent processing the same search again
CDRSCTI	The amount of time that dynamic CDRSCs are saved after the last session has ended
CINDXSIZ	The maximum size of the ISTCIT and ISTCONVT index tables
DIRSIZE	The maximum number of dynamic resources that VTAM stores in the APPN directory database
DIRTIME	The amount of time that a dynamic resource can go unused before it is deleted from the APPN directory database
HNTSIZE	The maximum size of the host node table used to find element addresses
HPRPST	The maximum time that VTAM tries a path switch before ending a connection.
HSRTSIZE	The size of the SRT directory for the network containing the VTAM host node
IOINT	The interval after which the VTAM operator is notified of outstanding I/O requests

Table 59. Start Options That Affect Performance and Tuning (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=PERFTUNE command.

Start Option	Description
MAXLOCAT	The maximum threshhold value to be reached before VTAM suspends sending APPN search requests to an adjacent node.
IOPURGE	The interval after which outstanding I/O requests are assumed to be lost and recovery steps are taken
MAXLURU	The maximum RU length for LU-LU sessions
MIHTMOUT	The interval in which you can adjust the missing interrupt handling interval for channel-to-channel communication
MXSAWBUF	The maximum number of buffers to use for session awareness data
MXSSCPRU	The maximum RU length for all SSCP sessions
NCPBUFSZ	The RU size used for loading or dumping a remote NCP
NSRTSIZE	Provides an override of OSRTSIZE for a maximum of nine network identifiers.
NUMTREES	The maximum number of APPN routing trees that can be saved
OSIEVENT	Specifies whether VTAM topology notifications are generated for CMIP services events
OSITOPO	Specifies whether the VTAM topology agent reports all types of cross domain resources (CDRSCs) to CMIP manager application programs.
OSRTSIZE	The size of the SRT directory for networks other than the VTAM host node's network
PDTRCBUF	The number of buffers to build for a session monitor request
PIUMAXDS	A value used to calculate the maximum buffer limit for PIU data space
PSRETRY	The interval between attempts to switch a rapid transport protocol (RTP) session to a better path
SAWMAXDS	A value used to calculate the maximum buffer limit for SAW data space
SAWMXQPK	A value used to calculate the number of SAW buffers to queue before packing the buffers
SONLIM	The maximum percentage of I/O buffers that can be used for session outage notification RUs
SRCHRED	Indicates whether this node enables eliminating and reducing searches for unreachable resources.
SRCOUNT	The number of search requests that fail before VTAM performs another resource discovery search
SRTIMER	The amount of time VTAM does not conduct searches for an unavailable resource
UPDDELAY	Specifies the maximum time the VTAM topology agent waits between checking the list of resources in transient states (such as PENDING ACTIVE and PENDING INACTIVE) to determine whether the resources have been in transient states long enough to be sent to the topology manager

Table 59. Start Options That Affect Performance and Tuning (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=PERFTUNE command.

Start Option	Description
VFYRED	Specifies whether the node should attempt to perform resource verification reduction
VFYREDTI	Specifies the node-wide maximum time interval that a resource's location is not verified
VTAMEAS	The size of the table that contains information about sessions with an application program

APPN Start Options

Table 60. Start Options That Define APPN Characteristics. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=APPNCHAR command.

Start Option	Description
APPNCOS	The APPN class of service that is used if the requested class of service cannot be found
BN	Whether this node is to provide extended border node function
BNDYN	Indicates how nodes are added dynamically to adjacent cluster routing lists
BNORD	Indicates the order VTAM performs cross-subnet searches
CDSERVR	Whether this node is a central directory server
CDSREFER	Specifies how many of the nearest central directory servers are used for Locate searches and central resource registration
CONNTYPE	Whether a LEN connection or an APPN connection is established with type 2.1 PUs
CPCP	The types of connections on which CP-CP sessions are allowed with adjacent nodes
DIRSIZE	The maximum number of dynamic resources that VTAM stores in the APPN directory database
DIRTIME	The amount of time that a dynamic resource can go unused before it is deleted from the APPN directory database
DYNADJCP	Whether dynamic entries are created in ISTADJCP for adjacent APPN nodes
DUPDEFS	Whether APPN should continue searching for the target resource when the resource has been found but is not active.
HPR	Specifies the level of high performance routing (HPR) support provided by VTAM
HPRNCPBF	Specifies whether HPR can be used although it may cause session data to travel through an NCP twice.
HPRPST	The maximum time that VTAM tries a path switch before ending a connection.
INITDB	Whether to load the APPN directory database, topology database, neither, or both when VTAM is started
IOPURGE	The interval after which outstanding I/O requests are assumed to be lost and recovery steps are taken
MAXLOCAT	The maximum threshhold value to be reached before VTAM suspends sending APPN search requests to an adjacent node.
NETID	The name of the network containing the host

Table 60. Start Options That Define APPN Characteristics (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=APPNCHAR command.

Start Option	Description
NNSPREF	Specifies the name of the preferred network node server for this end node.
NODETYPE	The APPN node type (network node or end node)
NUMTREES	The maximum number of APPN routing trees that can be saved
PSRETRY	The interval between attempts to switch a rapid transport protocol (RTP) session to a better path
PSWEIGHT	The weight comparison between an old route and a newly calculated route before a path switch is attempted.
RESUSAGE	The number of times an APPN resource (node or TG) can be used for routing before it is considered overused
ROUTERES	The relative amount of resistance to having this APPN node perform intermediate session routing
SAVERSCV	Indicates whether or not Route Selection Control Vectors (RSCVs) for future sessions are to be saved for display purposes following the completion of session activation.
SECLVLCP	Whether VTAM uses basic or enhanced protocol for session-level security verification during activation of LU 6.2 sessions involving APPN control points
SORDER	The order in which to search APPN and subarea networks when a network search request is received from a subarea network
SRCHRED	Indicates whether this node enables eliminating and reducing searches for unreachable resources
SRCOUNT	The number of search requests that fail before VTAM performs another resource discovery search
SRTIMER	The amount of time VTAM does not conduct searches for an unavailable resource
SSCPNAME	The name of the VTAM SSCP or CP
SSEARCH	Whether to search a subarea network when a network search request is received from an APPN network
SNVC	Maximum number of networks that can be searched for a resource by this border node.
VERIFYCP	Whether VTAM performs LU-LU session-level verification during activation of LU 6.2 sessions involving APPN control points
VFYRED	Specifies whether the node should attempt to perform resource verification reduction
VFYREDTI	Specifies the node-wide maximum time interval that a resource's location is not verified
VRTG	Whether virtual-route-based transmission groups connections are requested when SSCP-SSCP session is established
VRTGCPCP	Whether CP-CP sessions are supported over the virtual-route-based transmission group

Functions From Zappable Constants

Table 61. Start Options That Were Zappable Constants. These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=ZAPCON command.

Start Option	Description
ASIRFMSG	The SSCPs issue messages IST890I and IST896I when an autologon session initiation request fails
BSCTMOUT	The interval which two general poll failures are allowed for BSC 3270 terminals
CINDXSIZ	The maximum size of the ISTCIT and ISTCONVT index tables
ESIRFMSG	The SSCPs issue messages IST891I, IST892I, and IST893I when a session initiation request fails and extended sense data exists
FSIRFMSG	The SSCPs issue messages IST894I, IST895I, IST1704I, and IST1705I when a session initiation request fails because VTAM is unable to locate the destination LU by using an adjacent SSCP table
HNTSIZE	The maximum size of the host node table used to find element addresses
HSRTSIZE	The size of the SRT directory for the network containing the VTAM host node
INOPDUMP	Whether to dump VTAM for certain INOP conditions
IOMSGLIM	The maximum number of IST530I and IST532I message pairs that are written for individual subareas by the IOPD facility
IRNSTRGE	The maximum amount of VTAM private storage that can be used for host intermediate routing node (IRN) transmissions
MAXLURU	The maximum RU length for LU-LU sessions
MAXSSCPS	The maximum number of SSCPs that are searched on a specific path for the destination SSCP before alternate paths are tried
MIHTMOUT	The interval which you can adjust the missing interrupt handling interval for channel-to-channel communication
MXSAWBUF	The maximum number of buffers to use for session awareness data
MXSSCPRU	The maximum RU length for all SSCP sessions
MXSUBNUM	The maximum subarea number supported by a network to which a VTAM host is interconnected
NCPBUFSZ	The RU size used for loading or dumping a remote NCP
OSRTSIZE	The size of the SRT directory for networks other than the VTAM host node's network
PDTRCBUF	The number of buffers to build for a session monitor request
PIUMAXDS	A value used to calculate the maximum buffer limit for PIU data space
PLUALMSG	Whether VTAM is to suppress duplicate session setup failure messages associated with LOGAPPL failures to the same PLU
PSSTRACE	Whether to record IRB and SRB entries in the VTAM internal trace table when the PSS trace option is in effect
SAWMAXDS	A value used to calculate the maximum buffer limit for SAW data space

Table 61. Start Options That Were Zappable Constants (continued). These start options can be displayed as a group with the DISPLAY VTAMOPTS,FUNCTION=ZAPCON command.

Start Option	Description
SAWMXQPK	A value used to calculate the number of SAW buffers to queue before packing the buffers
SDLCMDRS	Which SDLC miscellaneous data records are to be recorded in LOGREC
SIRFMSG	The SSCPs in which to issue IST663I, IST664I, IST889I, and subordinate messages when a session initiation request fails
SLUALMSG	Whether to suppress messages for session setup failures that occur when the SLU is not enabled
SNAPREQ	The number of requests for VTAM buffers between snapshot dumps
TRANSLAT	Which names are to be translated by alias name translation
SSDTMOUT	The amount of time that a switched PU can go without a session before the connection is dropped
VTAMEAS	The size of the table that contains information about sessions with an application program

Descriptions of Start Options

The remainder of this chapter describes the start options in alphabetical order. Information given about each start option includes the function of the option, the default, if any, and any restrictions on the option's use.

Note: For some start options, there is a table following the syntax diagram which shows a dependency, or a value range, or both. If there is not a table following a syntax diagram, then there is not a dependency or a value range.

ALSREQ

```
►►─┬─ALSREQ=NO──────────┬──────────────────────────────────────────►◄
   │                    │
   └─ALSREQ=──┬─YES─┬──┘
              └─NO──┘
```

Specifies whether adjacent link station names (or an appropriate matching name) must appear in the default adjacent link station list of the independent LU at a session request.

If adjacent link station names are required, only session requests over a matching adjacent link station in the list are allowed. For a LEN adjacent link station, this means the names must match. For an APPN adjacent link station, it means that another APPN adjacent link station (real or ISTAPNPU) is found in the list. If adjacent link station names are not required, any adjacent link station can receive a session request for this LU. Furthermore, the link station name is added to the list if it is LEN, or ISTAPNPU is added to the list if it is APPN.

You can change the value of ALSREQ with the MODIFY VTAMOPTS command while VTAM is running.

The value you specify on the ALSREQ start option provides the default for the ALSREQ operand on the CDRSC definition statement if you do not specify it.

ALSREQ=NO
Specifies that an adjacent link station name need not match an entry in the default link station list of the independent LU.

ALSREQ=YES
Specifies that an adjacent link station name must match an entry in the default link station list of the independent LU.

APPNCOS

```
                (1)
   ┌APPNCOS=NONE──────────────┐
►►─┤                          ├──────────────────────────────────────────►◄
   └APPNCOS=class-of-service_name┘
```

Notes:

1 APPNCOS is meaningful only if the NODETYPE start option is also used.

Specifies the APPN class of service to be used if a requested class of service cannot be found in the topology and routing services class-of-service database. See “IBM-Supplied APPN Classes of Service” on page 680 for the names of the seven classes of service and “APPN Class-of-Service Definitions” on page 668 for information on implementing a user-defined class of service.

If a requested class of service cannot be found, the value you specify in APPNCOS is substituted for it, and there is a possibility that the characteristics of the substitute class of service are not the ones you intended for the route.

For example, a secure class of service might have been intended, but the substitution provided in APPNCOS might offer a class of service that uses unsecured links.

You can change the value of APPNCOS with the MODIFY VTAMOPTS command while VTAM is running.

ASIRFMSG

```
   ┌ASIRFMSG=OLUSSCP──────┐
►►─┤                      ├──────────────────────────────────────────►◄
   └ASIRFMSG=──┬ALLSSCP─┬─┘
               ├OLUSSCP─┤
               └NONE────┘
```

Controls the display of messages IST890I and IST896I. These messages are issued when an autologon session initiation request fails.

You can change the value of ASIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

ASIRFMSG=ALLSSCP
Specifies that messages are issued in all SSCPs.

ASIRFMSG=OLUSSCP
Specifies that messages are issued only in the origin logical unit (OLU) SSCP.

ASIRFMSG=NONE
Specifies that no messages are issued in any SSCP.

ASYDE

```
┌─ASYDE=TERM──────┐
►►──┤                ├──────────────────────────────────────────►◄
└─ASYDE=──┬─KEEP─┬──┘
          └─TERM─┘
```

Specifies whether to keep or terminate the session between the device and its application when VTAM receives an asynchronous device end from a local non-SNA device. For example, an asynchronous device end is caused when the device is powered off, or, for a printer, when the cover is lifted.

This option applies to all locally attached non-SNA devices except printers. See also “FEATUR2” on page 196 for a description of FEATUR2=PRINTR or NOPRINTR for a LOCAL device.

ASYDE=KEEP
Specifies that the session is to remain up. VTAM generates an exception request for the application, and the application determines whether to end the session.

ASYDE=TERM
Specifies that VTAM unconditionally terminates the session.

AUTHLEN

```
┌─AUTHLEN=YES─────┐
►►──┤                ├──────────────────────────────────────────►◄
└─AUTHLEN=──┬─NO───┬──┘
            └─YES──┘
```

Indicates whether VTAM will pass the transmission priority specified from one LEN node to another LEN node.

AUTHLEN=NO
Indicates that the transmission priority specified by the entry LEN node will not be passed by VTAM; instead VTAM will set the transmission priority field to medium priority.

AUTHLEN=YES
Indicates that VTAM will forward the specified transmission priority from one LEN node to another.

AUTORTRY

```
   ┌─AUTORTRY=AUTOCAP──────┐
►►─┤                       ├──────────────────────────────────────────►◄
   └─AUTORTRY=──┬─AUTOCAP─┬─┘
                ├─CDRM────┤
                ├─ALL─────┤
                └─NONE────┘
```

Specifies which adjacent node activation will cause a retry of pending automatic logon requests.

Note: For an NN-EN configuration where the controlling PLU resides in the EN, the NNs might not receive the controlling PLU's registration at the time of the session attempt. To ensure autologon redrives, code the AUTOTI start option.

You can change the value of AUTORTRY with the MODIFY VTAMOPTS command while VTAM is running.

AUTORTRY=AUTOCAP
Specifies that automatic logon requests are retried only when an adjacent CDRM or an adjacent CP which supports automatic logon is activated.

AUTORTRY=CDRM
Specifies that automatic logon requests are retried only when an adjacent CDRM is activated.

AUTORTRY=ALL
Specifies that automatic logon requests are retried when any adjacent CP or CDRM is activated.

AUTORTRY=NONE
Specifies that automatic logon requests are not retried when any adjacent node is activated.

AUTOTI

```
   ┌─AUTOTI=0──────────────┐
►►─┤                       ├──────────────────────────────────────────►◄
   └─AUTOTI=time_period────┘
```

range: 0–7 days (or equivalent seconds, minutes, or hours)

Specifies how often pending automatic logon requests owned by this host are retried. AUTOTI is specified in seconds (S), minutes (M), hours (H), or days (D).

If you specify a number equal to or greater than 1 but less than 60 with an S, or without any suffix, AUTOTI defaults to 60 seconds. A value of 0 specifies that an automatic logon request is not retried during the period that an application (acting as the primary logical unit) is unavailable. When AUTORTRY≠NONE, VTAM reissues the automatic logon request when the application becomes available.

You can change the value of AUTOTI with the MODIFY VTAMOPTS command while VTAM is running.

BN

```
                (1)
   ┌─BN=NO──────┐
►►─┤            ├──────────────────────────────────────────────────►◄
   └─BN=──┬─NO──┬┘
          └─YES─┘
```

Notes:

1 BN is meaningful only if the NODETYPE=NN start option is also used.

Specifies whether this node is to provide extended border node function.

BN=NO
Specifies that this node does not provide extended border node function.

BN=YES
Specifies that this node provides extended border node function.

BNDYN

```
                      (1)
   ┌─BNDYN=LIMITED──────┐
►►─┤                    ├──────────────────────────────────────────►◄
   └─BNDYN=──┬─LIMITED──┬┘
             ├─NONE─────┤
             └─FULL─────┘
```

Notes:

1 BNDYN is meaningful only if the BN=YES start option is also used.

Defines how VTAM will add nodes dynamically to the adjacent cluster routing list when searching for a resource whose NETID matches the NETID specified on the NETID statement.

You can change the value of BNDYN with the MODIFY VTAMOPTS command while VTAM is running.

BNDYN=NONE
Specifies that no entries will be added to the routing table except those explicitly identified by the NEXTCP definition statements.

BNDYN=LIMITED
Specifies that all active border nodes in the native subnetwork, as well as active border nodes and peripheral network nodes in nonnative subnetworks, attached to this node are automatically added to the routing list if either of the following conditions is met:

- BNs and nonnative network nodes that VTAM learns about whose network ID matches the network ID of the DLU.
- BNs or nonnative network nodes that have previously sent a Locate that carried the network ID of the DLU.

BNDYN=FULL
Specifies that all active border nodes in the native subnetwork as well as active

border nodes and peripheral network nodes in nonnative subnetworks attached to this node are automatically added to the routing list.

BNORD

```
                    (1)
  ┌─BNORD=PRIORITY──────┐
►►┼──────────────────────┼──────────────────────────────────────────►◄
  └─BNORD=──┬─PRIORITY─┬─┘
            └─DEFINED──┘
```

Notes:

1 BNORD is meaningful only if the BN=YES start option is also used.

Specifies the order in which VTAM will perform cross-subnet searches.

You can change the value of BNORD with the MODIFY VTAMOPTS command while VTAM is running.

BNORD=DEFINED
Specifies that searches are done in the order that you define border nodes and nonnative network nodes.

Note: BNORD=DEFINED only applies to defined routing entries. Dynamic entries are reordered in priority order and placed after all defined entries.

BNORD=PRIORITY
Specifies that search preference is given to nodes for which the most recent search was successful and to nodes whose NETID matches the DLU's NETID.

BSCMDRS

```
  ┌─BSCMDRS=(STATS,INOPS)──────────────────────┐
►►┼────────────────────────────────────────────┼──────────────────────►◄
  └─BSCMDRS=──┬─STATS──────────────────────────┤
              ├─NOSTATS────────────────────────┤
              ├─(,INOPS)───────────────────────┤
              ├─(,NOINOPS)─────────────────────┤
              └─(──┬─STATS──┬─,INOPS──┬──────┬─)─┤
                   │        └─,NOINOPS┘      │
                   └─NOSTATS──┬─,INOPS──┬────┘
                              └─,NOINOPS┘
```

Specifies whether VTAM records subarea BSC inoperative and statistical miscellaneous data records (MDRs) in LOGREC.

Note: If you do not record MDRs, and a problem occurs in a BSC device, problem diagnosis can be difficult because there are no MDRs in LOGREC.

You can change the value of BSCMDRS with the MODIFY VTAMOPTS command while VTAM is running.

BSCMDRS=STATS
Specifies that VTAM will record the BSC statistical MDRs in LOGREC.

BSCMDRS=NOSTATS
Specifies that VTAM will not record BSC statistical MDRs in LOGREC.

BSCMDRS=INOPS
Specifies that VTAM will record the BSC inoperative MDRs in LOGREC.

BSCMDRS=NOINOPS
Specifies that VTAM will not record BSC inoperative MDRs in LOGREC.

BSCTMOUT

```
►►─┬─BSCTMOUT=286──────────────┬──────────────────────────────────────►◄
   └─BSCTMOUT=units_of_time────┘
```

range: 0–2147483647

Specifies an interval during which VTAM tracks the number of general poll failures that occur for a BSC 3270 terminal. If two failures occur within this interval, VTAM treats this as an unrecoverable failure of the terminal and deactivates it.

There are 1.048576 seconds in each *units_of_time*. If 0 is specified, the terminal is not deactivated as a result of general poll failures.

Buffer Pool

```
►►──┬──────────────────────────────────────────┬──────────────────────►◄
    └─poolname=──┬─()──────────────────────────┬─┘
                 └─┤ Buffer Pool Values ├──────┘
```

Buffer Pool Values:

```
├──(──┬──────────┬──,──┬──────────┬──,──┬──────────┬──,──┬───┬──,──┬──────────┬──,──────►
      └─baseno───┘     └─bufsize───┘     └─slowpt───┘     └─F─┘     └─xpanno───┘

►──┬──────────┬──,──┬──────────┬──)──────────────────────────────────────────────────┤
   └─xpanpt───┘     └─xpanlim───┘
```

Allows you to modify the default buffer pool attributes for the various pools (both basic allocation and dynamic allocation values). Buffer pools are used by VTAM for holding data or building control blocks. Refer to “Buffer Pool Default Values” on page 570 for information on the IBM-supplied buffer pool values.

For more information on buffer pools, types of buffer pools, and guidelines for basic buffer allocation and dynamic buffer allocation, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

poolname
The name of the fixed-length buffer pool to which these options apply.

poolname can be the full buffer pool name, the buffer pool ID, or the abbreviation (for example, either SFBUF, SF00, or SF, respectively, for specifying a small buffer pool in fixed storage).

Refer to Table 62 for a list of the buffer pool names and a description of each buffer pool's function. A complete description of each buffer pool is located in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Table 62. VTAM Buffer Pools

Poolname	Pool ID	Abbreviation	Use
APBUF	AP00	AP	Used for device related CIBs
BSBUF	BS00	BS	Provides storage for the boundary type 2.1, type 2, and type 1 peripheral node session control blocks
CRA4BUF	CRA4	CRA4	Used for error recovery
CRA8BUF	CRA8	CRA8	Used for error recovery
CRPLBUF	CRPL	CR	Contains the RPL-copy pool
IOBUF	IO00	IO	Used for input/output data
LFBUF	LF00	LF	Provides buffers needed for each active application program and TSO user logged on
LPBUF	LP00	LP	Used for error recovery
SFBUF	SF00	SF	Contains application program information and LU control blocks
SPBUF	SP00	SP™	Used for large message (LMPEO) requests
TIBUF	TI00	TI	Contains control information to support HPDT services
XDBUF	XD00	XD	Provides storage for the I/O buffer during XID contact processing to peripheral nodes

baseno

range: 1–32767
dependencies: *bufsize*

Specifies the base number of buffers in the pool. Do not code commas in the value specified.

Notes:

1. The value for *baseno* should contain the maximum number of buffers that will fit on each page. If *baseno* is smaller than the maximum number of buffers per page, VTAM adjusts the value to the maximum value.
2. Storage for buffer pools is allocated by page. After you know the number of buffers per page (for the given *bufsize*), you can determine the maximum *baseno* for your environment.
3. Because storage is allocated by page increments, the amount of buffer storage reserved per page might be less than the size of a page. This storage difference must be considered when determining the value for *xpanlim*.

bufsize

range: 256–3992 bytes
dependencies: *baseno*, IOBUF pool only

Specifies the number of bytes assigned to each buffer in the pool. If you specify a value less than 256, it is changed to 256. If an odd value is specified, it is rounded down to the closest even number.

Notes:

1. The *bufsize* value specified must allow individual buffers to begin on a doubleword boundary and not expand over a page boundary. See "Specifying IOBUF Buffer Values" on page 571 for specifying IO buffer pool *bufsize* values.
2. The value for *bufsize* does not include header information added to each buffer by VTAM. See "Specifying IOBUF Buffer Values" on page 571 for determining the actual size of each buffer.
3. The only buffer pool for which you can change the *bufsize* is IOBUF. You may code the default values for the other buffer pools. If you specify a value other than the default value, a message will be issued and initialization will continue with VTAM using the default values.
4. Some local SNA controllers, such as 3174, require an even number and will not operate if *bufsize* is an odd value. To accommodate this requirement, a specification of an odd *bufsize* value will be rounded down (–1) to the next even value. Refer to the appropriate device description manual for other resource types.

slowpt

dependencies: *baseno*

Provides a set of buffers reserved for priority requests (for example, read from a channel-attached device). When the number of available buffers in this buffer pool is equal to or less than this number, buffers are allocated only for priority requests, and normal requests are queued or rejected. This is called slowdown mode.

F Specifies that the storage for this buffer pool is to be page-fixed. F has meaning only for buffer pools that are normally in pageable storage, but it can be coded for buffer pools that are located in fixed storage by default.

Notes:

1. To use the *xpanno* and *xpanpt* operands, but not the F operand, code either double commas or 0 between the *slowpt* operand and the *xpanno* operand. For example, coding either:

   ```
   LPBUF=(baseno,bufsize,slowpt,,xpanno,xpanpt,xpanlim)
    or
   LPBUF=(baseno,bufsize,slowpt,0,xpanno,xpanpt,xpanlim)
   ```

   Causes the buffer pool's default attribute to be used.
2. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* to determine which buffer pools can be pageable.

xpanno

dependencies: *bufsize*
range: 0–32767

Specifies the number of buffers that VTAM acquires when expanding the buffer pool. This value is rounded to the number of buffers that fills the nearest whole page of storage, however the 32767 maximum values cannot be exceeded. For

such a case, *xpanno* is rounded down to the largest number less than or equal to 32767 that will completely fill a whole number of pages.

Notes:

1. If you code 0, dynamic buffering is not performed and the buffer pool never expands. If you do not code *xpanno*, the IBM-supplied defaults are used.
2. The use of an *xpanno* value that is too small for a pool affects run time performance rather than storage utilization. This is because VTAM will be frequently expanding and/or contracting the pool. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for dynamic expansion guidelines.

xpanpt

range: (*slowpt* + 1)–[(*baseno* - *minval*) - 1]

Specifies the number of buffers that set the expansion point for this buffer pool. When the number of available buffers in the buffer pool falls to a value that is equal to or less than *xpanpt*, VTAM expands the buffer pool by the number of buffers defined by *xpanno*.

Notes:

1. Specify a value for *xpanpt* so that the buffer pool is not expanded during VTAM initialization.
2. The value of *xpanpt* must be greater than the value of *slowpt*, but less than the difference between *baseno* and *minval*. *minval* is the minimum number of buffers required to open the VTAM ACB and start VTAM. The *minval* values are listed with the IBM-supplied defaults (see “Buffer Pool Default Values” on page 570).

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information and examples of dynamic buffer allocation.

xpanlim

dependencies: IOBUF pool only, *xpanno*
range: 0–2097152 Kb

Specifies the total size, in Kb, for this buffer pool including the base allocation and expansions.

VTAM transforms the value of *xpanlim* into the number of buffers represented by that limit.

Notes:

1. If you code 0, or if you do not specify an *xpanlim*, buffer expansion is not limited, provided the value for *xpanno* is nonzero.
2. If *xpanlim* is less than the number of bytes in the basic buffer pool allocation, no expansion is performed.

When a successful expansion of the IO buffer pool brings the total size of the pool to 80 percent (or more) of the expansion limit specified, VTAM scans the IO buffer pool for sessions that are using more than 10 percent of the current size of the buffer pool. Any session using more than 10 percent of the pool receives an IST930I message. This message contains the session partner

names as well as the percentage of the pool being used by the session. This happens for each expansion in which the size of the pool is 80 percent or greater of the specified expansion limit.

After VTAM has determined that a session is using more than 10 percent of the IO buffer pool, it checks whether the percentage is greater than or equal to the value specified for HOTIOTRM. If so, VTAM ends the session and any other session between the same resource pair to prevent CSA from being exhausted and possibly causing an outage.

Buffer Pool Default Values: Table 63 shows the IBM-supplied values for the buffer pools.

Notes:

1. The *bufsize* value specified is not the actual length of a buffer; VTAM adds additional storage to each buffer for buffer management. The amount of storage added is the same for all buffers except for the IOBUF buffer pool.
   - To estimate storage needs for all buffer pools except IOBUF:
     a. Add 16 bytes to the buffer size.
     b. Round *bufsize* up to a doubleword (multiple of 8).
   - For IOBUF (no encryption specified):
     a. Add 91 bytes to the *bufsize* value.
     b. Round *bufsize* up to a doubleword (multiple of 8).
   - For IOBUF, if YES, 24, or 31 is specified on the ENCRYPTN start option:
     a. Add 98 bytes to the *bufsize* value.
     b. Round *bufsize* up to a doubleword (multiple of 8).
2. *minval* is the minimum number of buffers required to open the VTAM ACB and start VTAM. The *minval* value is a constant. You cannot change this value.
3. The minimum value for IOBUF bufsize that will be used by VTAM is 256. If a value less than 256 is specified, then it will be over-ridden to 256 during VTAM initialization.

```
D NET,BFRUSE,BUFFER=IO
IST920I IO00  BUFFSIZE 331
```

or

```
D NET,BFRUSE,BUFFER=IO
IST920I IO00  BUFFSIZE 338
```

Table 63. IBM-Supplied VTAM Buffer Pool Values

Buffer Pool Name	baseno	bufsize	slowpt	xpanno	xpanpt	xpanlim	minval
APBUF	56	56	2	56	3	-	-
BSBUF	24	298	0	12	12	-	-
CRA4BUF	12	4080	0	6	2	-	-
CRA8BUF	12	8176	0	6	2	-	-
CRPLBUF	60	144	0	60	29	-	-
IOBUF	51	256	5	1	6	0	-
LFBUF	30	120	30	1	1	-	-
LPBUF	12	2032	0	6	2	-	4
SFBUF	21	176	0	21	1	-	5
SPBUF	21	176	0	21	1	-	-

Table 63. IBM-Supplied VTAM Buffer Pool Values (continued)

Buffer Pool Name	baseno	bufsize	slowpt	xpanno	xpanpt	xpanlim	minval
TIBUF	360	632	0	60	120	-	-
XDBUF	5	697	0	5	4	-	-

Overriding Buffer Pool Values: All buffer pool values that you specify override the previously specified values, while buffer pools that you do not code default to the IBM-supplied values.

In the following example, LPBUF values are specified in the default start option list. When VTAM is started, the system operator selects start option list ST, which contains new LPBUF values which will override the previous values. VTAM processes these values in the order shown below. For more details on processing, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Overriding Buffer Pool Example: In this example, the operator enters LIST=ST. Note that the double commas denote values that are not coded.

```
IBM default            LPBUF=(9,1344,0,,1,2)
ATCSTR00               LPBUF=(128,,1,,0,0)
ATCSTRST (LIST=ST)     LPBUF=(8,,0,F,,)
```

The resultant values for LPBUF are: (8,1344,0,F,0,0).

Specifying IOBUF Buffer Values: If you want to maximize real storage usage, choose a buffer size (*bufsize*) that fits efficiently onto a page. The following calculations can be made to determine an optimum value for your environment. When selecting a value, you need to know the average PIU size to have a starting point for *bufsize*.

Note: For type 2 local SNA devices, the actual data size is 16 bytes less than IOBFSIZ.

To calculate how many buffers fit on a page (no encryption specified):

1. Determine the *bufsize* value.
2. Add 95 bytes to the specified *bufsize* value.
3. Round the sum up to a doubleword boundary.
4. Divide 4096 by the value.
5. The quotient indicates the number of buffers that can fit on 1 page (for the given *bufsize* value).

   The remainder indicates the number of bytes that are not used on each page. However, these unused bytes are taken into consideration for the *xpanlim* value that you specify. Keep this number as low as possible.

If the encryption start option is specified (ENCRYPTN = YES, 24, or 31), add 98 to the *bufsize* value instead of 95.

Table 64 on page 572 reflects the IOBUF formula and can be used when coding *bufsize*. For greatest efficiency, choose the high number in the range, because the higher number has a smaller remainder.

Note: The minimum *bufsize* used by VTAM is 256. If a *bufsize* of less than 256 is specified, VTAM overrides the value with a value of 256 and displays no message indicating this has occurred. D NET,BFRUSE,BUFFER=IO will, however, show the resulting change.

Table 64. IOBUF Buffer Size and Number of Buffers per Page

Number of buffers per page	Range of buffer sizes for MVS(with 4k paging)	Range of buffer sizes for MVS using data encryption facility (with 4k paging)
1	4004–1958	3998–1950
2	1956–1270	1950–1262
3	1268–934	1262–926
4	832–726	926–718
5	724–590	718–582
6	588–494	582–486
7	492–422	486–414
8	420–358	414–350
9	356–318	350–310
10	316–278	310–270
11	276-256	270-256

CACHETI

```
►►─┬─CACHETI=8──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─CACHETI=number_of_minutes──┘
```

range: 0–1440 minutes

Specifies the minimum number of minutes that procedure-correlation identifiers (PCIDs) are cached after VTAM processes a network search request. Cached PCIDs are used to recognize previously processed search requests, thereby preventing duplicate processing.

Note that if you set the CACHETI value too low, duplicate searching can result. Conversely, if you set the value too high, storage use can increase due to VTAM retaining the control blocks used to save the PCIDs.

CDRDYN

```
►►─┬─CDRDYN=YES──────────┬──────────────────────────────────────────►◄
   │                     │
   └─CDRDYN=─┬─NO──┬─────┘
             └─YES─┘
```

Specifies whether the *host* is authorized to dynamically define CDRSC representations of cross-domain, cross-network, or APPN resources when a session request is received from or sent to an adjacent SSCP or CP.

Note: For V4R1 and V4R2 MVS, CDRDYN does not have a default.

You can change the value of CDRDYN with the MODIFY VTAMOPTS command while VTAM is running.

Notes:

1. For information specific to cross-domain sessions, see “CDRDYN” on page 120.
2. The CDRDYN start option applies only to the host CDRM. The CDRDYN start option value, or a MODIFY VTAMOPTS value, overrides only the CDRDYN value on the *hosts'* CDRM definition statement.
3. Using the CDRDYN start option instead of the CDRDYN operand allows greater flexibility with the DYNLU start option.

CDRDYN=NO
Causes the failure of any cross-domain, cross-network, or APPN session request for a cross-domain, cross-network, or APPN resource that has no predefined CDRSC entry.

CDRDYN=YES
Authorizes dynamic definition of cross-domain, cross-network, or APPN resources by this host, eliminating the need to predefine certain CDRSCs. Both origin logical units and destination logical units can be dynamically defined.

CDRSCTI

```
►►─┬─CDRSCTI=8M──────────────┬──────────────────────────────────────────►◄
   │                         │
   └─CDRSCTI=timeout_value───┘
```

dependencies:	applies only to dynamic CDRSCs
range:	1 second–7 days

Specifies the minimum amount of time for which the system retains a dynamically defined cross-domain resource after the last session with the resource has been terminated.

There are several considerations when specifying a value for CDRSCTI for an interchange node. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for details.

For a description of the retention time for APPN resources, see “DIRTIME” on page 582.

timeout_value can be specified in the following notation, where *n* is an integer:

n or nS
Specifies the number of seconds a resource is retained

nM Specifies the number of minutes a resource is retained

nH Specifies the number of hours a resource is retained

nD Specifies the number of days a resource is retained

CDSERVR

```
                 (1)
   ┌─CDSERVR=NO──────┐
►►─┤                 ├──────────────────────────────────────────────────►◄
   └─CDSERVR=─┬─YES─┬─┘
              └─NO──┘
```

Notes:

1 CDSERVR is meaningful only if the NODETYPE=NN start option is also used.

Specifies whether this network node is a central directory server. If connected to ISTGENERIC, the value of CDSERVR is set to yes.

CDSERVR=NO
Specifies that this network node is not a central directory server.

CDSERVR=YES
Specifies that this network node is a central directory server.

CDSREFER

```
                 (1)
   ┌─CDSREFER=1──────────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   └─CDSREFER=number_of_CDSs─┘
```

Notes:

1 CDSREFER is meaningful only if the NODETYPE=NN and CDSERVR=NO start options are also used.

range: 0–65535

Specifies how many of the nearest central directory servers (CDS) are used for Locate searches and central resource registration. “Nearest” refers to the least-weight CP-CP session paths to a CDS. If CDSREFER=0, then all CDSs in the network are used.

You can change the value of CDSREFER with the MODIFY VTAMOPTS command while VTAM is running.

CINDXSIZ

```
   ┌─CINDXSIZ=8176────────┐
►►─┤                      ├──────────────────────────────────────────►◄
   └─CINDXSIZ=table_size──┘
```

range: 28–16777199 bytes

Specifies the maximum size of the ISTCIT and ISTCONVT index tables.

CMPMIPS

```
                (1)
   ┌─CMPMIPS=100──────────────┐
►►─┤                          ├──────────────────────────────────────────►◄
   └─CMPMIPS=compression_ratio─┘
```

Notes:

1 The CMPMIPS start option is meaningful only if the value for CMPVTAM is greater than 1.

range: 0–100

Used by VTAM to determine the amount of time the adaptive compression tables are in adaptive mode versus being static:

- When an adaptive data compression table is in adaptive mode, CMPMIPS is used to determine the amount of time the compression table remains in adaptive mode before becoming static. When a compression table is static, no new entries are made to the table.
- When an adaptive data compression table is static, CMPMIPS is used to calculate the lowest compression efficiency rate that a session must maintain to keep the compression tables static.

The higher the value specified, the greater the amount of time spent in adaptive mode. The lower the value, the sooner a compression table becomes static.

You can change the value of CMPMIPS with the MODIFY VTAMOPTS command while VTAM is running.

Notes:

1. Adaptive mode uses CPU cycles; a static table does not use CPU cycles. However, when a table is static, compression efficiency decreases.
2. If CMPMIPS=0, no compression processing is performed.
3. If CMPMIPS=100, adaptive mode compression is continuous and the compression table is never static.
4. CMPMIPS=50 provides the most effective beginning balance between compression efficiency and CPU usage.

CMPVTAM

```
   ┌─CMPVTAM=0────────────────┐
►►─┤                          ├──────────────────────────────────────────►◄
   └─CMPVTAM=compression_level─┘
```

range: 0–4

Specifies the maximum compression level allowed for sessions involving the host's application programs. Specify one of the following values for *compression_level*:

0 No compression
1 Run-length encoding (RLE)
2 Small adaptive compression table
3 Medium adaptive compression table
4 Large adaptive compression table

You can change the value of CMPVTAM with the MODIFY VTAMOPTS command while VTAM is running.

CNMTAB

```
►►─┬─CNMTAB=ISTMGC00──────────┬──────────────────────────────────────────►◄
   │                          │
   └─CNMTAB=cnm_table_name────┘
```

Specifies the name of the communication network management (CNM) routing table to be used by this VTAM to route unsolicited request units for further processing. The value specified for *cnm_table_name* is the 1–8 character name of a user-defined CNM routing table.

For more information on coding a user-defined CNM routing table, refer to *OS/390 SecureWay Communications Server: SNA Customization*.

COLD and WARM

```
►►─┬─COLD─┬──────────────────────────────────────────────────────────────►◄
   │      │
   └─WARM─┘
```

Defines the status to which the VTAM configuration restart facility restores each major node identified by the CONFIG start option when VTAM was last executing.

COLD
Instructs VTAM to restore each major node to its initial status as defined by the user. VTAM issues the equivalent of:

```
VARY NET,ACT,ID=major_node_name,SCOPE=U
```

for each major node identified by the CONFIG start option.

WARM
Instructs VTAM to restore the major node to its status prior to VTAM's last termination. VTAM issues the equivalent of:

```
VARY NET,ACT,ID=major_node_name,WARM
```

for each major node identified by the CONFIG start option.

CONFIG

```
►►─┬─CONFIG=00────────────────┬──────────────────────────────────────────►◄
   │                          │
   └─CONFIG=─┬─file_id───────┬┘
             └─file_name─────┘
```

Contains a list of definition members in the VTAM definition library that are to be activated when VTAM is started. For more information on configuration lists, refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

CONFIG=00
Files a user-defined configuration list under the name ATCCON00.

VTAM always uses this predefined list unless a system operator uses the CONFIG option to select another list, or another list of start options specifies a different CONFIG start option. If a default configuration list does not exist when VTAM is started, an error message is sent to the system or domain operator. VTAM initialization continues without the configuration list.

Note: The configuration list named by CONFIG replaces the default configuration list. This differs from start option list processing where LIST, if coded, is merged with the 00 list.

CONFIG=*file_id*
Provides 1–2 alphanumeric characters that specify the file (or member) in the VTAM definition library that contains a list of major nodes that are activated when VTAM is started.

File the list of major nodes under the name ATCCON*xx*.

CONFIG=*file_name*
The 3–8 character file name of the configuration restart VSAM file. The file name must have been coded in the NODELST option during VTAM's execution prior to the failure or termination of VTAM.

This file contains a list of the major nodes that were active and a list of the dynamic reconfiguration data set (DRDS) files and sets of PATH statements that were in effect at the time of failure or termination of VTAM.

Include a DD statement using this data definition name in the VTAM start procedure.

CONNTYPE

```
                    (1)
  ┌CONNTYPE=APPN──────┐
►►┼──────────────────┼──────────────────────────────────────────►◄
  └CONNTYPE=──┬─APPN─┬┘
              └─LEN──┘
```

Notes:

1 CONNTYPE is meaningful only if the NODETYPE start option is also used.

Specifies for an APPN PU (type 2.1) whether a connection from it is established as a LEN connection or attempted as an APPN connection. CONNTYPE applies to end nodes and network nodes.

You can change the value of CONNTYPE with the MODIFY VTAMOPTS command while VTAM is running.

CONNTYPE=APPN
Specifies that the connection is attempted as an APPN connection. When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

CONNTYPE=LEN
Specifies that the connection is established as a LEN connection. CONNTYPE=LEN specifies that this connection does not support CP-CP sessions and CP name change, and that VTAM does not check for duplicate CPNAMEs.

CPCDRSC

```
  ┌─CPCDRSC=NO──────┐
►►─┤                ├──────────────────────────────────────────────►◄
  └─CPCDRSC=─┬─YES─┬─┘
             └─NO──┘
```

Specifies whether VTAM allows applications or LUs to initiate and establish sessions to a dynamic LEN CP ILU. A dynamic LEN CP ILU is an ILU with a resource name that is the same as its adjacent link station CPNAME.

CPCDRSC=NO
Specifies that VTAM does not allow applications or LUs to initiate and establish sessions with dynamic LEN CP ILUs. However, the LEN CP ILU can initiate and establish the session successfully.

CPCDRSC=YES
Specifies that VTAM allows applications and LUs to initiate and establish sessions with a dynamic LEN CP ILU.

CPCP

```
                  (1)
  ┌─CPCP=LEASED──────┐
►►─┤                ├──────────────────────────────────────────────►◄
  └─CPCP=─┬─YES────┬─┘
          ├─NO─────┤
          ├─LEASED─┤
          └─SWITCHED┘
```

Notes:

1 CPCP is meaningful only if the NODETYPE start option is also used.

Specifies whether an APPN node supports CP-CP sessions with an adjacent node.

You can change the value of CPCP with the MODIFY VTAMOPTS command while VTAM is running.

CPCP=YES
Specifies that CP-CP sessions on all connections are supported.

CPCP=NO
Specifies that CP-CP sessions are not supported for any connection, leased or switched.

CPCP=LEASED
Specifies that CP-CP connections are supported for leased connections, but not supported for switched connections.

CPCP=SWITCHED
Specifies that CP-CP sessions are supported for switched connections, but not supported for leased connections.

CSALIMIT

```
►►─┬─CSALIMIT=0──────┬──────────────────────────────────────────────────►◄
   └─CSALIMIT=value──┘
```

range: 0–2048M (2 gigabytes)

Specifies the maximum amount of common service area (CSA) that can be used by VTAM. This can be expressed as a number followed by a K (which represents 1024 bytes), or a number followed by an M (which represents 1048576 bytes). If you do not code M or K, K is assumed.

You can change the value of CSALIMIT with the MODIFY VTAMOPTS command or the MODIFY CSALIMIT command while VTAM is running. The F operand of this command allows you to force the value specified to be used as the CSA usage limit even though VTAM is currently using more CSA. If the operator forces a lower limit to take effect, the operator is prevented from using VTAM commands until the CSA usage falls below the specified limit, because CSA storage is needed to process all VTAM operator commands except HALT CANCEL. If the CSA usage does not fall below the specified CSA limit, VTAM must be canceled and restarted with a more appropriate CSALIMIT value specified.

Notes:

1. If the number you code is greater than the system CSA limit (90% of CSA + ECSA defined in the system) CSALIMIT is set to the system CSA limit (which is the default). Refer to the *OS/390 SecureWay Communications Server: SNA Operation*, DISPLAY BFRUSE, for information on displaying the system CSA limit.
2. If the limit you code is reached, errors are likely to occur. If LPBUF cannot be expanded, VTAM could enter an interlock condition. Therefore, if you code CSALIMIT, define LPBUF so that it does not have to expand. Other possible consequences are lost messages or failures when sessions are initiated or terminated.
3. Specifying CSALIMIT does not affect the storage used by the communications storage manager (CSM).

CSALIMIT=0
Specifies that the maximum amount of CSA storage to be used by VTAM is the system CSA limit. Refer to the *OS/390 SecureWay Communications Server: SNA Operation*, DISPLAY BFRUSE, to find out what the system CSA limit is for your system.

CSALIMIT=value
Specifies the maximum amount of CSA storage to be used by VTAM. *value* can be specified in any of the following forms:

nK
n is the number of 1024-byte increments that is the maximum amount of CSA that can be used by VTAM. The value of *n* is rounded up to the next multiple of 4.

qM
q is the number of 1-megabyte increments that is the maximum amount of CSA that can be used by VTAM.

CSA24

```
►►─┬─CSA24=0─────────┬──────────────────────────────────────────────────►◄
   └─CSA24=value─────┘
```

range: 0–16M

Specifies the maximum amount of 24-bit addressable common service area (CSA) that can be used by VTAM. This can be expressed as a number followed by a K (which represents 1024 bytes), or a number followed by an M (which represents 1048576 bytes). If you do not code M or K, K is assumed.

If VTAM does not explicitly request the desired storage location and the operating system returns CSA addressable storage below the 16 MB line (24-bit addressable storage), the CSA24 limitations do not apply.

You can change the value of CSA24 with the MODIFY VTAMOPTS command or the MODIFY CSALIMIT command while VTAM is running. The F operand of these commands allows you to force the value specified to be used as the CSA usage limit.

CSA24=0
Specifies that no limit is enforced on the amount of 24-bit addressable CSA used by VTAM.

CSA24=value
Specifies the maximum amount of CSA storage to be used by VTAM. *value* can be specified in any of the following forms:

nK
n is the number of 1024-byte increments that is the maximum amount of 24-bit addressable CSA that can be used by VTAM. The value of *n* is rounded up to the next multiple of 4.

qM
q is the number of 1 MB increments that is the maximum amount of 24-bit addressable CSA that can be used by VTAM.

DATEFORM

```
►►─┬─DATEFORM=MDY────────┬──────────────────────────────────────────────►◄
   └─DATEFORM=─┬─MDY─┬──┘
               ├─DMY─┤
               └─YMD─┘
```

Specifies the format of the start date for an automatic scheduled IPL.

Note: DATEFORM or DATEFRM is allowed.

If the year is omitted, the current year is used. The MODIFY LOAD command is rejected if the value entered for the date is any of the following:

- Not valid
- Greater than 90 days from the current date or time
- Prior to current date or time

Note: You will encounter a MOSS clock problem if you set the IPL to occur after 15 days. For MOSS clock accuracy, the IPL should be less than or equal to 15 days. Otherwise, you can set the IPL up to 90 days in advance.

DATEFORM=DMY
Represents the date format DD/MM/YY. The valid delimiters are a slash (/) or a period (.).

DATEFORM=MDY
Represents the date format MM/DD/YY. The valid delimiters are a slash (/) or a dash (-).

DATEFORM=YMD
Represents the date format YY/MM/DD. The valid delimiters are a slash (/), a period (.), or dash (-).

DIALRTRY

```
      ┌─DIALRTRY=YES──────┐
►►────┤                   ├──────────────────────────────────────────────►◄
      └─DIALRTRY=──┬─NO──┬─┘
                   └─YES─┘
```

Specifies whether VTAM should continue searching for a switched resource after a dial–out failure. Searching should be allowed to continue if the switched resource is also defined (an possibly already active) on another VTAM. Searching should not be allowed to continue if the switched resource is only defined on this VTAM. You can change the value of DIALRTRY with the MODIFY VTAMOPTS command while VTAM is running.

DIALRTRY=NO
Specifies that VTAM should not continue searching for a switched resource after a dial–out failure.

DIALRTRY=YES
Specifies that VTAM should continue searching for a switched resource after a dial–out failure.

DIRSIZE

```
                (1)
      ┌─DIRSIZE=0──────────────────────┐
►►────┤                                ├──────────────────────────────────►◄
      └─DIRSIZE=number_of_resources────┘
```

Notes:

1 DIRSIZE is meaningful only if the NODETYPE=NN start option is also used.

range: 0, or 1000–2147483647

Specifies the maximum number of dynamic APPN resources that VTAM stores in the directory services database. Once the number specified is reached, storage from the oldest dynamic resources is freed and reused.

0 specifies that no limit is enforced for the number of dynamic APPN resources.

You can change the value of DIRSIZE with the MODIFY VTAMOPTS command while VTAM is running.

DIRTIME

```
                (1)
   ┌─DIRTIME=8D──────────┐
►►─┤                     ├──────────────────────────────────────────────►◄
   └─DIRTIME=time_period─┘
```

Notes:

1 DIRTIME is meaningful only if the NODETYPE=NN start option is also used.

dependencies:	applies only to APPN resources
range:	25 hours–2 years (730D)

Specifies the amount of time that a dynamic APPN resource can remain unused in the directory services database before VTAM deletes it. Dynamic resources older than the time specified are deleted.

You can specify the time in seconds (S), hours (H), or days (D).

DIRTIME is valid only when NODETYPE=NN, and applies only to dynamic APPN resources. See “CDRSCTI” on page 573 for a description of the retention time for subarea resources.

You can change the value of DIRTIME with the MODIFY VTAMOPTS command while VTAM is running.

DISCNTIM

```
   ┌─DISCNTIM=(15,0)──────────────────────┐
►►─┤                                      ├──────────────────────────────►◄
   └─DISCNTIM=(time_period1,time_period2)─┘
```

time_period1:	5–65535 seconds
time_period2:	0–255 seconds

Specifies the amount of time VTAM delays deactivation of an SSCP-PU session when there are no LU-LU session requests.

For Subarea and Non-HPR APPN Sessions: DISCNTIM is valid only for PU types 2 and 2.1 that have DISCNT=DELAY specified on the PU definition statement.

time_period1 specifies the amount of time VTAM delays deactivation of an SSCP-PU session after all LU-LU sessions terminate.

time_period2 specifies the amount of time after establishment of an SSCP-PU session, VTAM delays deactivation of the session if no LU-LU sessions have been established. When 0 is specified, VTAM will not deactivate the line if no LU-LU session is established after the line is connected.

You can change the value of DISCNTIM with the MODIFY VTAMOPTS command while VTAM is running.

DLRORDER

```
►►─┬─DLRORDER=STATNID──────────┬──────────────────────────────────────────►◄
   │                           │
   └─DLRORDER=──┬─STATNID─┬────┘
                └─CPNAME──┘
```

Specifies the way VTAM locates a DLUR PU. If you do not code DLRORDER, VTAM searches for a DLUR PU by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes) first. If VTAM does not find the PU by the station identifier (STATNID), it searches for the DLUR PU by CPNAME.

You can change the value of DLRORDER with the MODIFY VTAMOPTS command while VTAM is running.

DLRORDER=CPNAME
Specifies that VTAM searches for a DLUR PU by the CPNAME first.

DLRORDER=STATNID
Specifies that VTAM searches for a DLUR PU by the station identifier first (default).

Note: For regular (non-DLUR) switched PUs, see the SWNORDER start option.

DLRTCB

```
►►─┬─DLRTCB=32──────────────────────────┬──────────────────────────────────►◄
   │                                    │
   └─DLRTCB=number_of_task_control_blocks┘
```

range: 1–32767

Specifies the largest number of task control blocks (TCBs) used by VTAM for dump-load-restart subtasks and file I/O services. If you code 0, VTAM uses the default value of 32.

The DLRTCB start option causes control blocks to be built below the 16-megabyte line. Therefore, the availability of storage might limit the value that you can code, and conversely, the value you code might impact the availability of the storage. Also, if several dump-load-restart subtasks are active at the same time, the value you code for this start option might affect performance.

If VTAM is initialized as an APPN network node (NODETYPE=NN), then it is recommended to specify at least 3 for DLRTCB. Two TCBs are needed to process directory services data sets and topology and routing services data sets, and one TCB is needed for other dump-load-restart functions.

A value less than 3 can cause VTAM initialization to stall, or suspend a session initiation for a function requiring a TCB until a TCB becomes available. For example, if DLRTCB is specified as 1, and directory services gets the TCB, topology and routing services will enter a wait state that will never be resolved because it requested a TCB that is not available.

DSPLYDEF

```
                 (1)
   ┌─DSPLYDEF=100──────────────┐
►►─┼───────────────────────────┼──────────────────────────────────────────►◄
   └─DSPLYDEF=number_of_messages─┘
```

Notes:

1 If the DSPLYMAX start option value is less than 100, that value is the default for DSPLYDEF.

dependencies:	none
range:	1–*value_of_DSPLYMAX*

Limits the number of messages displayed when the following commands are issued without the MAX operand:

DISPLAY ADJSSCPS	DISPLAY APPLS
DISPLAY CDRMS	DISPLAY CDRSCS
DISPLAY CLSTRS	DISPLAY DIRECTRY,ID=*.*name*
DISPLAY EXIT	DISPLAY GROUPS
DISPLAY LINES	DISPLAY LUGROUPS,SCOPE=ALL
DISPLAY MAJNODES	DISPLAY PATHTAB
DISPLAY PENDING	DISPLAY RSCLIST
DISPLAY SESSIONS,LIST=ALL	DISPLAY STATIONS
DISPLAY STATS	DISPLAY STORUSE
DISPLAY TABLE,SCOPE=ALL	DISPLAY TERMS
DISPLAY TGPS	DISPLAY TRL
DISPLAY USERVAR	

You can change the value of DSPLYDEF with the MODIFY VTAMOPTS command while VTAM is running.

DSPLYMAX

```
   ┌─DSPLYMAX=65535────────────┐
►►─┼───────────────────────────┼──────────────────────────────────────────►◄
   └─DSPLYMAX=number_of_messages─┘
```

dependencies:	none
range:	1–65535

Limits the value that can be specified for the DSPLYDEF start option and for the MAX operand on the following commands:

DISPLAY ADJSSCPS
DISPLAY APPLS
DISPLAY CDRMS
DISPLAY CDRSCS
DISPLAY CLSTRS
DISPLAY DIRECTRY,ID=*.*name*
DISPLAY EXIT
DISPLAY GROUPS
DISPLAY LINES
DISPLAY LUGROUPS,SCOPE=ALL
DISPLAY MAJNODES
DISPLAY PATHTAB
DISPLAY PENDING
DISPLAY RSCLIST
DISPLAY SESSIONS,LIST=ALL
DISPLAY STATIONS
DISPLAY STATS
DISPLAY STORUSE
DISPLAY TABLE,SCOPE=ALL
DISPLAY TERMS
DISPLAY TGPS
DISPLAY TRL
DISPLAY USERVAR

DSPLYWLD

```
►►─┬─DSPLYWLD=FULLWILD──────┬──────────────────────────────────────────►◄
   │                        │
   └─DSPLYWLD=──┬─FULLWILD─┬─┘
                ├─NOWILD───┤
                ├─OPERONLY─┤
                └─POAONLY──┘
```

Specifies whether wildcards are permitted in certain DISPLAY commands.

You can change the value of DSPLYWLD with the MODIFY VTAMOPTS command while VTAM is running.

DSPLYWLD=FULLWILD
Specifies that wildcards are permitted in DISPLAY commands from all network operators. Wildcards are permitted in DISPLAY commands from program operator applications whose APPL definition statement indicates DSPLYWLD=YES.

DSPLYWLD=NOWILD
Specifies that wildcards are not permitted in any DISPLAY commands.

DSPLYWLD=OPERONLY
Specifies that wildcards are permitted in DISPLAY commands from the network operator, but not from program operator applications.

DSPLYWLD=POAONLY
Specifies that wildcards are permitted in DISPLAY commands from program operator applications whose APPL definition statement indicates DSPLYWLD=YES. Wildcards are not permitted from the network operator.

DUPDEFS

```
►►─┬─DUPDEFS=ALL────────┬──────────────────────────────────────────►◄
   │                    │
   └─DUPDEFS=──┬─ALL───┬─┘
               ├─NONE──┤
               ├─APPL──┤
               └─DEPLU─┘
```

Specifies whether APPN should continue searching for the target resource when the resource has been found but is not active. The search is continued because it may find a duplicate definition of the resource that is not active. This start option should be set on the VTAM that owns the resource or on the VTAM acting as interchange node for resources located in subarea. You can change the value of DUPDEFS with the MODIFY VTAMOPTS command while VTAM is running.

DUPDEFS=ALL
Specifies that resources may be defined on more than one node in the network. If this node receives an APPN search for a resource that is not active, APPN should continue searching for the resource because an active instance may reside somewhere else in the network.

DUPDEFS=NONE
Specifies that APPN should not continue searching for a resource when it is found but not active. This value would be coded if there are not duplicate definitions in the network.

DUPDEFS=APPL
Specifies that applications may be defined on more than one node in the network. If this node receives an APPN search for an application that is not active, APPN should continue searching for the application because an active instance may reside somewhere else in the network.

DUPDEFS=DEPLU
Specifies that dependent LUs may be defined on more than one node in the network. If this node receives an APPN search for a dependent LU that is not active, APPN should continue searching for the dependent LU because an active instance may reside somewhere else in the network.

DYNADJCP

```
                    (1)
  ┌─DYNADJCP=YES──────┐
►►─┤                   ├──────────────────────────────────────────────────►◄
  └─DYNADJCP=──┬─YES─┬─┘
               └─NO──┘
```

Notes:

1 DYNADJCP is meaningful only if the NODETYPE start option is also used.

Specifies whether adjacent control point (ADJCP) minor nodes are allowed to be created dynamically and placed in ISTADJCP. The ADJCP minor node provides control and management of connections between this node and an adjacent APPN type 2.1 node. This option can be overridden at the connection level (that is, on a link station).

By specifying this option as NO and overriding its value on specific link stations, you can limit connections to a specific set of CPs over some link stations, and allow connections to any CP over other link stations.

DYNADJCP=NO
ADJCP minor nodes are not created dynamically. When a connection is established, the adjacent control point minor node must be defined and active or the connection is discontinued.

DYNADJCP=YES
ADJCP minor nodes are created as needed. Because some VTAM LEN

implementations do not have an ADJCP major node, this option defaults to YES to provide compatibility for all type 2.1 connections.

DYNASSCP

```
►►─┬─DYNASSCP=YES──────────┬──────────────────────────────────────────►◄
   │                       │
   └─DYNASSCP=──┬─YES─┬────┘
                └─NO──┘
```

Specifies whether VTAM dynamically routes session establishment requests to all active adjacent SSCPs if no appropriate adjacent SSCP table is defined.

Note: If you have coded a node to function as an end node or network node (NODETYPE=EN or NN, and HOSTSA is not coded), then DYNASSCP should not be coded because it has no effect. If you code it, error messages will be issued, but the node will come up.

DYNASSCP=NO
Specifies that VTAM does not perform dynamic session establishment request routing. If you have not coded the appropriate adjacent SSCP tables and have not used the SSCPDYN start option, or have not specified the CDRM owner, session establishments might fail.

DYNASSCP=YES
Specifies that VTAM determines the adjacent SSCPs when routing a session establishment request across a domain or across a network. If VTAM does not locate an appropriate adjacent SSCP table, it dynamically routes the session establishment request to all active adjacent SSCPs until the correct SSCP is found.

DYNDLGMD

```
►►─┬─DYNDLGMD=NONE─────────────┬──────────────────────────────────────►◄
   │                           │
   └─DYNDLGMD=──┬─logmode_name─┬─┘
                └─NONE─────────┘
```

Specifies the default logon mode name used for dynamic cross-domain resources when the resource is acting as the secondary logical unit (SLU) and a logon mode name is not otherwise provided

You can change the DYNDLGMD value with the MODIFY VTAMOPTS command while VTAM is running.

You can also change the default logon mode name used for a specific dynamic cross-domain resource with the MODIFY DEFAULTS or MODIFY RESOURCE command.

DYNDLGMD=*logmode_name*
Specifies the default logon mode name for dynamic cross-domain resources.

DYNDLGMD=NONE
Specifies that there is no default logon mode name for dynamic cross-domain resources.

DYNHPPFX

```
►►──┬──────────────────────────────────┬──────────────────────────────────────►◄
    │                      (1)         │
    └─DYNHPPFX=──prefix─────────────────┘
```

Notes:

1 Two character prefix.

If DYNHPPFX is not specified, then dynamic PUs created for HPR will be prefixed with the three characters, CNR. If DYNHPPFX=cc where cc is the desired two character prefix, then dynamic PUs created for HPR will be prefixed with the two characters, cc. In these two cases, respectively, the dynamic PU names will have the format of CNRxxxxx or ccxxxxxx.

DYNHPPFX=*prefix*
Specifies a two character prefix that is used in creating the dynamic PUs created for HPR. A unique value will then be generated and concatenated to the defined prefix to create an eight character PU name.

DYNLU

```
    ┌─DYNLU=NO──────────┐
►►──┤                   ├──────────────────────────────────────────────────────►◄
    └─DYNLU=──┬─YES─┬───┘
              └─NO──┘
```

Allows you to dynamically allocate cross-domain resource (CDRSC) definitions for resources being treated as independent LUs by this VTAM. (The cross-domain resources might actually represent application programs, dependent LUs or independent LUs in some other node's domain.)

You can override this value at the connection level by coding DYNLU on specific adjacent link stations (PU definition statements). The adjacent link station determines the boundary function link that the resource uses to access the network.

Likewise, for an adjacent control point major node, you can override the value of the DYNLU start option by specifying a different value on the DYNLU operand on the ADJCP definition statement.

DYNLU=NO
Specifies that VTAM will not use dynamically allocated CDRSC representations for sessions over adjacent link stations.

If you code DYNLU=NO, you are required to define all resources that can be accessed over adjacent link stations. If you do not define a given resource, sessions to or from that resource will fail.

DYNLU=YES
Specifies that VTAM can use dynamically allocated CDRSC representations for sessions over adjacent link stations.

If you code DYNLU=YES, you are not required to predefine all resources that can be accessed over adjacent link stations.

For subarea nodes and APPN-only nodes (HOSTSA is not specified), DYNLU=YES is meaningful only when CDRDYN=YES is coded, as the start option (either specified as YES or by the default) or as the operand.

DYNPUPFX

```
►►─┬─DYNPUPFX=CN──────────────┬──────────────────────────────────────►◄
   └─DYNPUPFX=──prefix¹⁴──────┘
```

DYNPUPFX=*CN*
Specifies the default.

DYNPUPFX=*prefix*
Specifies a two character prefix that is used in creating the dynamic PUs. This prefix will be used when generating names for these dynamically built PUs. The prefix defined by the start option will be used if DYNPUPFX is not coded on the group statement for the line under which the PU is built. A unique value will then be generated and concatenated to the defined prefix to create an eight character PU name.

DYNVNPFX

```
►►─┬──────────────────────────┬──────────────────────────────────────►◄
   └─DYNVNPFX=──prefix¹⁴──────┘
```

If DYNVNPFX is not specified, then dynamic PUs created for connection network will be prefixed with the three characters, CNV. If DYNVNPFX=vv where vv is the desired two character prefix, then dynamic PUs created for connection network will be prefixed with the two characters, cc. In these two cases, respectively, the dynamic PU names will have the format of CNVxxxxx or vvxxxxxx.

DYNVNPFX=*prefix*
Specifies a two character prefix that is used in creating the dynamic PUs created for connection network. A unique value will then be generated and concatenated to the defined prefix to create an eight character PU name.

DYNMODTB

```
►►─┬─DYNMODTB=NONE────────────┬──────────────────────────────────────►◄
   └─DYNMODTB=──┬─table_name─┬─┘
                └─NONE───────┘
```

Specifies the name of a logon mode table used for dynamic cross-domain resources to correlate each logon mode name with a set of session parameters when the resource acts as the SLU.

You can change the DYNMODTB value with the MODIFY VTAMOPTS command or the MODIFY TABLE,OPTION=LOAD command while VTAM is running.

You can also override this value for a specific dynamic cross-domain resource with the MODIFY TABLE,OPTION=ASSOCIATE command while VTAM is running.

DYNMODTB=*table_name*
Specifies the name of the default logon mode table used for dynamic cross-domain resources. The name specified must be the name of a logon mode table created as described in “Logon Mode Table” on page 698.

DYNMODTB=NONE
Specifies that there is no default logon mode table for dynamic cross-domain resources.

ENCRPREF

```
►►─┬─ENCRPREF=NONE──────────┬──────────────────────────────────────────►◄
   │                        │
   └─ENCRPREF=──┬─NONE──┬───┘
                └─value─┘
```

Specifies the prefix to be used for common cryptographic architecture (CCA) key-encrypting key (KEK) labels.

You can change the ENCRPREF value with the MODIFY VTAMOPTS command while VTAM is running.

ENCRPREF=NONE
Specifies that no prefix is to be used for the common cryptographic architecture (CCA) key-encrypting key (KEK) labels.

Note: The default or the IBM Integrated Cryptographic Service Facility (ICSF) in CCA and the Transaction Security System (TSS) program product running in PCF mode is no prefix.

ENCRPREF=value
Specifies the 1–8-character prefix to be used for common cryptographic architecture (CCA) key-encrypting key (KEK) labels.

Note: Transaction Security System (TSS) program product running in CCA mode has a configration data set. The PCF-KEY-PREFIX command specifies the first key label token name (the first 1-8 characters of the key label). The default for PCF-KEY-PREFIX is $$CUSP$$.

ENCRYPTN

```
►►─┬─ENCRYPTN=YES───────┬──────────────────────────────────────────►◄
   │                    │
   └─ENCRYPTN=──┬─YES──┬─┘
                ├─NO───┤
                ├─CCA──┤
                ├─CUSP─┤
                ├─24───┤
                └─31───┘
```

Specifies that 24-bit capable or 31-bit capable cryptography is installed.

Note: If the ENCRYPTN start option is not specified, VTAM will determine if cryptography is active and the addressing mode it supports.

ENCRYPTN=YES
Specifies that VTAM supports cryptography and determines the addressing mode at VTAM initialization.

At initialization:

- If the currently active cryptographic product supports 24-bit addressing mode, VTAM supports cryptographic products that support 24-bit or 31-bit addressing mode and VTAM I/O buffers are placed in 24-bit storage.
- If the currently active cryptographic product supports 31-bit addressing mode, VTAM only supports cryptographic products that support 31-bit addressing mode and VTAM I/O buffers are placed in 31-bit storage.
- If no cryptographic product is currently active, VTAM only supports cryptographic products that support 31-bit addressing mode and VTAM I/O buffers are placed in 31-bit storage.

ENCRYPTN=NO
Specifies that VTAM does not support cryptography, regardless of whether or not a cryptographic product is currently active.

ENCRYPTN=CCA
Specifies that VTAM use a common cryptographic architecture (CCA) product to satisfy cryptography requests. If a CCA product is not available at start time the request is rejected and the session fails. VTAM supports this product using 24-bit or 31-bit addressing mode.

ENCRYPTN=CUSP
Specifies that VTAM use an IBM Cryptographic Unit Support (CUSP) program product to satisfy a request for cryptographic support. If a CUSP product is not available at start time, the request is rejected and the session fails. VTAM supports CUSP using 24-bit or 31-bit addressing mode.

ENCRYPTN=24
Specifies that VTAM supports cryptography in 24-bit mode. VTAM I/O buffers are placed in 24-bit storage. VTAM uses cryptographic products that support either 24-bit or 31-bit storage addresses.

ENCRYPTN=31
Specifies that VTAM supports cryptography in 31-bit mode. VTAM I/O buffers are placed in 31-bit storage. VTAM only uses cryptographic products that support 31-bit addressing mode. If VTAM receives a cryptography request and the only available cryptographic product supports 24-bit addressing mode, the request fails.

ENHADDR

```
►►─┬─ENHADDR=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─ENHADDR=──┬─NO──┬──┘
               └─YES─┘
```

Specifies whether VTAM can assign element addresses greater than 65,535. These addresses are assigned to resources establishing sessions within this subarea and as a base address to application programs within this subarea.

ENHADDR=NO
Specifies that you cannot use high-order element addresses for application programs.

If you specify ENHADDR=NO all element addresses that VTAM obtains for resources in this subarea come out of the 64k element pool. This includes each active (connectable) APPL (two if PARSESS=YES), each activated local LU and PU, each dynamically defined LU and PU (once connected). We must also obtain a session partner element address for each session in which we are the endpoint or intermediate routing point (unless the session path to that partner is over a FID_4 link - because, in this case, the element address will originate from another subarea's address space). APPN session partner element addresses come out of this subarea's 64K element address pool if you specify ENHADDR=NO. Finally, every SSCP_LU and SSCP_PU element address for DLUR/DLUS comes out of this 64K element address pool.

ENHADDR=YES
Specifies that you can use high-order element addresses for application programs.

If you specify ENHADDR=YES and have an application program that used the network-qualified address pair control vector X'15', refer to *OS/390 SecureWay Communications Server: SNA Migration*.

SNA PUs have an architected element address pool limit of 64k, however, VTAM has created an additional pool of address spaces whose total is 16348k. In addition, VTAM must still obtain an element address from the 64k pool if the session path, in or out of the host, is over a FID_4 link.

A SNA network address is in the form of a four byte subarea address and a two byte element address (as in 'ssssssss eeee'), however, only the two low-order bytes of the subarea address are actually used. The mechanism VTAM uses to create and access a different element address space is to use the two high-order bytes as an index field (as in iiiissss eeee).

ESIRFMSG

```
  ┌─ESIRFMSG=ALLSSCP──────┐
►►─┤                       ├──────────────────────────────────────────►◄
  └─ESIRFMSG=─┬─ALLSSCP─┬─┘
              ├─OLUSSCP─┤
              └─NONE────┘
```

Controls the display of IST891I, IST892I, and IST893I. These messages are issued when a session initiation request fails and extended sense data exists.

You can change the value of ESIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

ESIRFMSG=ALLSSCP
Specifies that messages are issued in all SSCPs.

ESIRFMSG=OLUSSCP
Specifies that messages are issued only in the origin logical unit (OLU) SSCP.

ESIRFMSG=NONE
Specifies that no messages are issued in any SSCP.

FLDTAB

```
   ┌─FLDTAB=ISTMSFLD──────────┐
►►─┤                          ├──────────────────────────────────────────►◄
   └─FLDTAB=──┬─table_name─┬──┘
              └─NONE───────┘
```

Specifies whether VTAM uses a message-flooding prevention table to suppress duplicate messages.

FLDTAB=ISTMSFLD
Indicates that VTAM uses the internal message flooding table.

FLDTAB=NONE
Indicates that VTAM will not suppress duplicate messages that are sent to the operator console or system hardcopy log.

FLDTAB=table_name
Indicates that VTAM uses a user-specified message-flooding prevention table to determine which duplicate messages are to be suppressed. *table_name* is the 1–8 character name of the table stored in the VTAM definition library.

If an attempt to load the specified table fails, VTAM issues an error message and uses the VTAM internal table.

If you specify ISTMSFLD as the table name, VTAM will use the IBM-supplied table located in the VTAM definition library.

FSIRFMSG

```
   ┌─FSIRFMSG=OLUSSCP─────────┐
►►─┤                          ├──────────────────────────────────────────►◄
   └─FSIRFMSG=──┬─ALLSSCP─┬──┘
                ├─OLUSSCP─┤
                └─NONE────┘
```

Controls the display of IST894I, IST895I, IST1704I, and IST1705I. These messages are issued when a session initiation request fails because trial and error routing, using an adjacent SSCP table, exhausted the table and was unable to locate the destination LU.

You can change the value of FSIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

FSIRFMSG=ALLSSCP
Specifies that messages are issued in all SSCPs.

FSIRFMSG=OLUSSCP
Specifies that messages are issued only in the origin logical unit (OLU) SSCP.

FSIRFMSG=NONE
Specifies that no messages are issued in any SSCP.

GWSSCP

```
  ┌─GWSSCP=YES──────┐
►►─┤                ├──────────────────────────────────────────────────►◄
  └─GWSSCP=──┬─YES─┬─┘
             └─NO──┘
```

Specifies that an SSCP can be a gateway SSCP. If you code GWSSCP, the SSCP can still process CDRSCs defined with the NETID operand.

GWSSCP=NO
Specifies that this SSCP cannot:

- Use the alias name translation facility or the alias selection function of the session management exit
- Reroute initiation requests, so it cannot be an intermediate SSCP in the session setup path
- Perform the gateway functions.

GWSSCP=YES
Allows the SSCP to:

- Use the alias name translation facility and the alias selection function of the session management exit routine.
- Reroute initiation requests.
- Perform gateway functions in a session setup path.

Note that GWSSCP=YES is meaningful only for subarea nodes and APPN interchange nodes (NODETYPE=NN and HOSTSA is coded).

If you code GWSSCP=YES on the receiving CDRM definition, NETID is required on the NETWORK definition statement of all external CDRMs.

If you have coded a node to function solely as an end node or network node (NODETYPE=EN or NN, and HOSTSA is not coded), then you should *not* code GWSSCP. If you code GWSSCP in this case (specifying either YES or NO), it is ignored and an informational message is issued.

If you have coded a node to function as a migration data host (NODETYPE=EN and HOSTSA is coded), then you should code GWSSCP=NO. If you code GWSSCP=YES, or allow it to default to YES, VTAM forces GWSSCP to NO and issues an informational message.

HNTSIZE

```
  ┌─HNTSIZE=4080─┐
►►─┤              ├──────────────────────────────────────────────────►◄
  └─HNTSIZE=4080─┘
```

range: 4080 bytes

Specifies the size of the host node table used to find element addresses. This start option is no longer user defined. If specified in the VTAM start list, the value will be forced to 4080 bytes.

HOSTPU

```
┌─HOSTPU=ISTPUS──────────────────┐
►►─┤                                ├──────────────────────────────────────►◄
└─HOSTPU=host_subarea_pu_name────┘
```

dependencies: see notes

Specifies the network name of the VTAM host subarea physical unit for this host. It is recommended that you code HOSTPU if you are using the NetView program or NLDM. If you do not code HOSTPU, VTAM uses ISTPUS as the host PU name.

Notes:

1. The name for HOSTPU must not be the same as the name for this CDRM or the name coded for the SSCPNAME start option for this SSCP. If the name for HOSTPU is the same as the name for SSCPNAME, HOSTPU is set to ISTPUS. This host PU name should be network-unique.
2. You are prompted for the correct name if a reserved name other than ISTPUS is specified on the HOSTPU start option or if the HOSTPU name and the SSCPNAME are the same.
3. If you do not use the default name (ISTPUS), the subarea can still be defined to the system as ISTPUS. The name that you use as the HOSTPU value is interchangeable with the name ISTPUS in VTAM commands.

HOSTSA

```
┌─HOSTSA=1───────────────────────┐
►►─┤                                │──────────────────────────────────────►◄
│                        (1)     │
└─HOSTSA=subarea_number──────────┘
```

Notes:

1 HOSTSA specifies the subarea number of this VTAM. If HOSTSA is not coded, then a default subarea number of 1 is used.

range: 1– to the maximum number of subareas allowed for the network

If HOSTSA= is coded and SACONNS= is not coded or SACONNS=YES is coded, then this VTAM will provide subarea function. If HOSTSA= is not coded, or HOSTSA= is coded and SACONNS=NO is coded, then this VTAM will not provide subarea function.

The combination of start options HOSTSA, NODETYPE, and SACONNS determines the overall node configuration and level of function support. The start options and resulting node configurations are intended to provide a smooth migration for existing subarea hosts.

See Table 65 on page 596 for a summary of the node type functions. When the node is configured as a network node or end node, attempts to activate FID4 (PU types 4 or 5) connections will fail. However, if the node is a migration data host or an interchange node, FID4 connections are allowed.

Table 65. Node Type Functional Summary

Node Type	NODETYPE	HOSTSA	SACONNS	CP-CP Sessions	SSCP-SSCP Sessions	NCP Ownership	Interchange Function
Subarea Node	(not coded)	*n* [1]	n/a	no	yes	yes	no
Interchange Node	NN	*n* [1]	(not coded) or YES	yes	yes	yes	yes
Migration Data Host	EN	*n* [1]	(not coded) or YES	yes	yes	no	no
Network Node	NN	(not coded)	n/a	yes	no	no	no
Network Node	NN	*n* [1]	NO	yes	no	no	no
End Node	EN	(not coded)	n/a	yes	no	no	no
End Node	EN	*n* [1]	NO	yes	no	no	no

Notes:

1. *n* represents a subarea host number.

Note: If you want a node to function solely as an end node or network node (NODETYPE=EN or NN), then HOSTSA should not be coded or SACONNS=NO must be coded.

HOSTSA must be less than or equal to the maximum number of subareas specified (MXSUBNUM start option). See “MXSUBNUM” on page 609 for more information on the MXSUBNUM start option.

Note: If you plan to incorporate a non-VTAM host, VTAM’s SSCPID and HOSTSA start options values should be coded so that *both* values are either higher or lower than the non-VTAM host.

HOTIOTRM

```
►►─┬─HOTIOTRM=0──────────────────────────┬──────────────────────────────►◄
   │                                     │
   └─HOTIOTRM=percentage_of_buffer_pool──┘
```

range: 0 or 10–99

Specifies the percentage of the current size of the IO buffer pool that a single session must have allocated to it to cause VTAM to take corrective action. Corrective action differs depending on the type of session involved:

- When the session type is SSCP-LU: inactivate the LU.
- When the session type is SSCP-PU: inactivate the PU.
- When the session type is LU-LU: terminate all sessions between the two logical units.

If you omit HOTIOTRM or specify 0, VTAM will not terminate sessions based on IO buffer pool usage.

You can change the value of HOTIOTRM with the MODIFY VTAMOPTS command while VTAM is running.

Notes:

1. It is very important that a proper expansion limit (XPANLIM) be specified on the IO buffer pool when HOTIOTRM is specified. An expansion limit allows VTAM to detect hot session(s) prior to complete CSA exhaustion. See the description of the IOBUF buffer pool for a complete description of the XPANLIM parameter.
2. Investigate past VTAM console logs for VTAM IST930I messages when there did not seem to be a problem. Specifying HOTIOTRM greater than the percentages in these past IST930I messages will cause VTAM to not terminate sessions that “normally” use large amounts of the IO buffer pool. Usually, these session have not had proper session level pacing values specified. You might wish to consider specifying session pacing that will reduce the amount of IO buffers used by these sessions.
3. If past console logs are not available, or no IST930I messages are found, specify a HOTIOTRM value of 50.

HPR

```
                (1)
   ┌─HPR=RTP──────────────┐
►►─┤                      ├──────────────────────────────────────────►◄
   └─HPR=──┬─ANR──────────┬┘
           ├─RTP──────────┤
           ├─NONE─────────┤
           ├─(RTP,ANR)────┤
           ├─(RTP,NONE)───┤
           └─(ANR,NONE)───┘
```

Notes:

1 HPR is meaningful only if NODETYPE is also used.

Specifies the level of high performance routing (HPR) support provided by VTAM.

HPR=ANR
Indicates this VTAM (and NCP in a composite network node) provides ANR-level HPR support. Coding HPR=YES on a CDRM or PU definition statement (or allowing it to default to YES) results in ANR-level HPR support over that connection.

HPR=ANR is not valid at an end node (NODETYPE=EN).

HPR=RTP
Indicates this VTAM provides RTP-level HPR support. Coding HPR=YES on a CDRM or PU definition statement (or allowing it to default to YES) results in RTP-level HPR support over that connection.

HPR=NONE
Indicates this VTAM (and NCP in a composite network node) provides no HPR support.

HPR=(RTP,ANR)
Indicates this VTAM provides RTP-level HPR support only if HPR=YES is coded on the CDRM or PU definition statement. Letting the HPR operand default on a CDRM or PU definition statement results in only ANR-level HPR support over that connection.

HPR=(RTP,ANR) is not valid at an end node (NODETYPE=EN).

HPR=(RTP,NONE)
Indicates this VTAM provides RTP-level HPR support only if HPR=YES is coded on the CDRM or PU definition statement. Letting the HPR operand default on a CDRM or PU definition statement results in no HPR support over that connection.

HPR=(ANR,NONE)
Indicates this VTAM provides ANR-level HPR support only if HPR=YES is coded on the CDRM or PU definition statement. Letting the HPR operand default on a CDRM or PU definition statement results in no HPR support over that connection.

HPR=(ANR,NONE) is not valid at an end node (NODETYPE=EN).

HPRNCPBF

```
►►─┬─HPRNCPBF=NO──────────┬──────────────────────────────────────────►◄
   │                      │
   └─HPRNCPBF=──┬─YES─┬───┘
                └─NO──┘
```

Specifies whether HPR can be used although it may cause session data to travel through an NCP twice. If HPR is used, session data may pass through the NCP as HPR data on the RTP connection AND also as non-HPR data.

HPRNCPBF=NO
Specifies HPR will not be used if session data might travel through an NCP twice.

HPRNCPBF=YES
Specifies that HPR can be used although it may cause session data to travel through an NCP twice.

When coding HPRNCPBF=YES, in general your considerations are possible decrease in network performance but better session availability.

You can change the value of HPRNCPBF with the MODIFY VTAMOPTS command while VTAM is running. If you change the value from YES to NO, no new RTP connections are activated that would cause session data to go through an NCP twice, but existing such RTP connections may in some cases be used.

See the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information.

HPRPST

```
►►─┬─HPRPST=(8M,4M,2M,1M)──────────────────────────────┬──────────────►◄
   │                                                   │
   └─HPRPST=(low_limit,med_limit,high_limit,network_limit)─┘
```

range: 5 secs–24 hours (low, medium, and high limits) 5 secs–2 minutes (network limit)

Specifies the maximum time that VTAM tries a path switch before ending a connection, when HPR=RTP. The value can be expressed as nS, nM, or nH. If you do not specify an S, M, or H, seconds are assumed. When a high performance routing RTP node determines that it is necessary to switch paths for the RTP logical connection, it starts a path switch timer.

You can specify different time limits for low, medium, high, and network transmission priority connections. For example, you might allow only a 20–second (20S) limit for interactive connections to stay up during a path switch, but allow an hour (1H) limit for batch connections to stay up while waiting for restoration of a path that was down for maintenance.

Though there are no strict rules for coding HPRPST, consider the following:

- Specifying too short an interval might not allow enough time for a path switch attempt to succeed and find a new route. This could be a problem, particularly when the network is congested.
- Specifying too long an interval might tie up storage and possibly leave an LU in a state where other LUs or applications could not communicate with it.
- The network transmission priority should be set at a substantially lower value than the other transmission priorities. This allows CP-CP sessions, the signals for which flow at network priority, to complete path switch before expiration of the path switch timer for an RTP connection carrying an LU-LU session.
- Although HPR connections established for some sessions with multinode persistent session (MNPS) applications might have a network transmission priority, these connections use the HPRPST value specified for the high transmission priority limit. See *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information about recovery of MNPS sessions.

You can change the value of HPRPST with the MODIFY VTAMOPTS command while VTAM is running. The changed HPRPST values do not affect existing RTP connections, but are used for any new RTP connections that are activated.

HSRTSIZE

```
►►─┬─HSRTSIZE=9973──────────────────────────┬──────────────────────────────────►◄
   └─HSRTSIZE=number_of_queue_pointers──────┘
```

range: 1–2097148

Specifies the number of queue pointers in the symbol resolution table (SRT) for the network containing the VTAM host node.

INITDB

```
                 (1)
     ┌─INITDB=ALL──────┐
►►───┤                 ├──────────────────────────────────────────────────►◄
     └─INITDB=─┬─ALL──┬─┘
               ├─DIR──┤
               ├─TOPO─┤
               └─NONE─┘
```

Notes:

1 INITDB is meaningful only if the NODETYPE=NN start option is also used.

Specifies whether the directory services and the topology and routing services databases are loaded when VTAM is started.

INITDB=ALL
Specifies that both databases are loaded.

INITDB=DIR
Specifies that only the directory services database is loaded.

INITDB=TOPO
Specifies that only the topology and routing services database is loaded.

INITDB=NONE
Specifies that neither database is loaded.

INOPDUMP

```
     ┌─INOPDUMP=OFF────┐
►►───┤                 ├──────────────────────────────────────────────────►◄
     └─INOPDUMP=─┬─ON───┬─┘
                 └─OFF──┘
```

Specifies whether VTAM dumps should be generated whenever VTAM detects certain channel-to-channel, local SNA, and high performance data transfer INOPs.

You can change the INOPDUMP value with the MODIFY VTAMOPTS command while VTAM is running.

INOPDUMP=OFF
Specifies that dumps are not taken.

INOPDUMP=ON
Specifies that dumps are taken.

IOINT

```
     ┌─IOINT=180───────────────┐
►►───┤                         ├──────────────────────────────────────────►◄
     └─IOINT=number_of_seconds─┘
```

range: 0–5366000

Specifies the number of seconds after which any outstanding response for most request units sent by VTAM is identified with a VTAM message.

Coding IOINT=0 deactivates the function. If you code a value less than 60, IOINT=60 (1 minute) is assumed.

You can change the value of IOINT with the MODIFY VTAMOPTS command or the MODIFY IOPD command while VTAM is running.

IOMSGLIM

```
►►─┬─IOMSGLIM=2147483647──────────────┬──────────────────────────────────────►◄
   └─IOMSGLIM=number_of_message_pairs─┘
```

range: 1–2147483647

Limits the number of pairs of IST530I and IST1436I messages that are written for individual subareas by the IOPD facility. If the number of pending I/O operations for a subarea is greater than the value specified, then the IOPD facility writes only one message pair for each type of pending I/O operation for that subarea instead of one message pair per pending operation. The default value ensures that a message pair is written for each pending I/O operation.

You can change the value of IOMSGLIM with the MODIFY VTAMOPTS command while VTAM is running.

IOPURGE

```
►►─┬─IOPURGE=0──────────────┬──────────────────────────────────────►◄
   └─IOPURGE=timeout_value─┘
```

range: 0, or 30 seconds–7 days (or equivalent value in seconds, minutes, or hours)

Specifies an interval after which outstanding I/O requests are purged. The types of outstanding I/O requests that are checked are:
- CDINIT requests
- Direct search list requests
- APPN search requests
- HPR route setup requests

Before this function (or when IOPURGE is set to zero), outstanding I/O requests could remain outstanding indefinitely. For example, when an intermediate host that is used in session routing goes down, it cannot respond to a session establishment request from an LU. The LU then remains hung as it waits indefinitely for a response. However, when an IOPURGE value expires, the outstanding request is canceled (a negative acknowledgement is sent) the LU is freed, and VTAM continues to search for a session establishment for the LU on an alternate path.

It is recommended that you set an IOPURGE value of three minutes or more, or set the value for twice as long as session setup usually takes.

In some circumstances, for example logging on a remote LU via a slow modem, session setup might take longer and the IOPURGE should be adjusted upward accordingly. In these circumstances, if you set IOPURGE too low, a session request is canceled before a response can be sent and received, and the session request will have to be repeated.

Setting too low a value could also generate unnecessary searches. For instance, using the above example of the intermediate routing host: the host might be slow to respond, rather than inactive, and setting too low a value for IOPURGE could cancel a request that would have completed and cause VTAM to continue to search for a session establishment.

If you specify a value less than 30 seconds, VTAM uses IOPURGE=30 (seconds).

You can change the value of IOPURGE with the MODIFY VTAMOPTS command or the MODIFY IOPURGE command while VTAM is running.

n or nS
Specifies the number of seconds that can pass before a response is assumed to be lost.

nM
Specifies the number of minutes that can pass before a response is assumed to be lost.

nH
Specifies the number of hours that can pass before a response is assumed to be lost.

nD
Specifies the number of days that can pass before a response is assumed to be lost.

0 Specifies that you do not want to use this function.

IPADDR

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └IPADDR=address────┘
```

Specifies the IP address at the local TCP/IP stack. If the local TCP/IP stack has multiple VIPA addresses, use IPADDR to specify which of those VIPA addresses you want other Enterprise Extender nodes to use in order to communicate with this host. If IPADDR is not specified, the default is to use the first VIPA address in the local TCP/IP stack's HOME list.

Note: IPADDR can be dynamically modified using MODIFY VTAMOPTS but the new value will not be used until all lines in the XCA major node used for Enterprise Extender are inactive. However, displays of VTAM start options will show the new value immediately. Any subsequent line activation from the Enterprise Extender XCA major node will make use of the new IPADDR value.

IRNSTRGE

```
   ┌─IRNSTRGE=0──────┐
►►─┤                 ├──────────────────────────────────────────────────►◄
   └─IRNSTRGE=size───┘
```

range: 1–2147483647 bytes

Defines the maximum size of the virtual area in VTAM storage that can save host intermediate routing node (IRN) transmissions. If zero (the default) is used, the amount of storage is unlimited.

ISTCOSDF

```
   ┌─ISTCOSDF=INDLU──────────────────┐
►►─┤                                 ├──────────────────────────────────►◄
   └─ISTCOSDF=──┬─ALL──────────────┬─┘
                ├─APPL─────────────┤
                ├─DEPLU────────────┤
                ├─INDLU────────────┤
                ├─NONE─────────────┤
                └─(──┬─APPL──────┬─)─┘
                     ├─DEPLU─────┤
                     └─INDLU─────┘
```

Specifies the resource types that can use the ISTCOSDF logmode entry. This entry is used when the logmode name specified for the session is not found.

You can change the value of ISTCOSDF with the MODIFY VTAMOPTS command while VTAM is running.

ISTCOSDF=ALL
Indicates that the ISTCOSDF logmode entry will be used by the following resources:
- Application programs
- Dependent LUs
- Independent LUs

ISTCOSDF=APPL
Indicates that ISTCOSDF is restricted to use by application programs.

ISTCOSDF=DEPLU
Indicates that ISTCOSDF is restricted to use by dependent LUs.

ISTCOSDF=INDLU
Indicates that ISTCOSDF is restricted to use by independent LUs.

ISTCOSDF=NONE
Indicates that the function is disabled.

LIMINTCP

```
►►─┤                                 ├──────────────────────────────────►◄
   └─LIMINTCP=number_of_seconds──────┘
```

dependencies: LIMRES=YES

range: 0–65535

Specifies the interval to retain a free SNASVCMG session on limited resources for a control point. The control point SNASVCMG session is used for some network management flows. LIMRES=YES must be coded on applicable GROUP, LINE, or PU definition statements for LIMINTCP to be effective.

If you do not code LIMINTCP, the limited resource function is not supported for network management flows. This function is supported only on network management flows using the CP SNASVCMG session. It is not supported on APPN CP sessions or dependent LU server sessions. Note that in a subarea environment, CP refers to the SSCP.

You can change the value of LIMINTCP with the MODIFY VTAMOPTS command while VTAM is running. However, though you can modify the value of LIMINTCP, there is not a way to dynamically turn off its function after VTAM has started.

LIST

```
                (1)
     ┌─LIST=00──────────────────┐
►►───┤                          ├──────────────────────────────────────────►◄
     └─LIST=start_option_list_id┘
```

Notes:

1 LIST can be entered by a VTAM operator only. If LIST is coded in an ATCSTR *-xx* file, it is considered to be an error and is ignored.

Specifies which list of predefined start options is used to start VTAM. Each ATCSTR*xx* list is filed in the VTAM definition library. You can choose any two alphanumeric characters (for *xx*) to identify user-defined lists of start options.

For more information on start option lists, refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

LISTBKUP

```
►►──┬──────────────────────────────────────────────────────────────────────►◄
    │                                        (1)
    └─LISTBKUP=──┬─start_option_list_id──┬──────┘
                 ├─DEFAULTS──────────────┤
                 └─PROMPT────────────────┘
```

Notes:

1 LISTBKUP can only be coded in a start option file. If you enter it on the START command or at an operator prompt, VTAM will ignore it.

Specifies what action is taken when VTAM detects an error in a start file.

LISTBKUP=*start_option_list_id*
Specifies the 2-letter start list identifier (ATCSTRyy) of another start file to process in place of the file in error. All options in the start file in error are set to the values before the file was processed and ATCSTRyy is processed.

If the backup file contains errors, LISTBKUP=yy is ignored to prevent possible looping. However, LISTBKUP=DEFAULTS or LISTBKUP=PROMPT is processed in a backup file.

LISTBKUP=DEFAULTS
Causes the options in the start file in error to be set to the values they had before the file was processed. VTAM then continues processing.

LISTBKUP=PROMPT
Causes all valid options in the start file to be set. The operator is prompted to enter overrides to the options that are not valid.

MAINTLVL

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─MAINTLVL=maintenance_level─┘
```

Specifies which maintenance level is running on a host. You can code up to eight characters to identify the maintenance level, and the identifier can be displayed by an operator.

If you do not specify a maintenance level identifier, blanks are displayed. Any characters except the equals sign (=), comma (,), and left and right parentheses are allowed.

MAXLOCAT

```
                  (1)
    ┌─MAXLOCAT=5000──────────────────────────┐
►►──┤                                        ├──────────────────────────────►◄
    └─MAXLOCAT=max_locate_congestion_threshhold─┘
```

Notes:

1 MAXLOCAT is meaningful only if NODETYPE is specified.

range: 100–65535

Specifies the maximum locate congestion threshhold value to be reached before VTAM suspends sending APPN search requests to an adjacent CP. The maximum locate congestion threshhold is determined by the following formula:

```
(queued requests for the adjacent CP) x
                 (number of seconds since last response from adjacent CP)
```

VTAM resumes sending APPN search requests to an adjacent CP when a minimum congestion threshhold is reached. The minimum threshhold is based on the number of queued requests waiting to be sent to the adjacent CP. When the queued requests are reduced to 10% of the number on the queue when the maximum locate congestion threshhold was reached, new APPN search requests are again sent to the adjacent CP.

If you specify a value less than 100, VTAM uses MAXLOCAT=100 as the MAXLOCAT value.

MAXLURU

```
►►─┬─MAXLURU=6144──────────┬──────────────────────────────────────────►◄
   └─MAXLURU=ru_length─────┘
```

range: 1–65535 bytes

Specifies the maximum length, in bytes, for request units (RUs) in LU-LU sessions that have a VTAM application as an endpoint.

You can change the value of MAXLURU with the MODIFY VTAMOPTS command while VTAM is running.

MAXSSCPS

```
►►─┬─MAXSSCPS=10──────────────┬──────────────────────────────────────────►◄
   └─MAXSSCPS=number_of_SSCPs─┘
```

range: 1–255

Specifies the maximum number of SSCPs on a specific path that are searched for the destination SSCP before alternate paths are tried. This number affects the rerouting of both session initiation requests (CDINIT) and direct search routing requests (INQUIRE APPSTAT).

You can change the value of MAXSSCPS with the MODIFY VTAMOPTS command while VTAM is running.

MAXSUBA

```
►►─┬─MAXSUBA=15──────────────────┬──────────────────────────────────────►◄
   └─MAXSUBA=maximum_subarea_number─┘
```

dependencies: required if ENA not supported
range: 1–255

Specifies the highest subarea value that can be assigned to any node within this network for communicating with NCPs and SSCPs that do not support extended network addressing (ENA).

This operand is required if any host or NCP in the network does not support ENA.

If you code MAXSUBA, the range of its values must be the same for all nodes in the same network.

Note: If you have coded a node to function solely as an end node or network node (NODETYPE=EN or NN), then MAXSUBA should not be coded because it has no effect. If you code it, error messages will be issued, but the node will come up.

The MAXSUBA value is rounded up to the next higher 2^n-1. For example, if MAXSUBA is coded as 10, the next higher 2^n-1 is 15. Therefore, VTAM uses a MAXSUBA value of 15.

In networks that have no pre-ENA nodes, or where you do not want to communicate with any such nodes, you do not need to code MAXSUBA.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for MAXSUBA usage considerations.

MIHTMOUT

```
      ┌─MIHTMOUT=1800──────────┐
►►────┤                        ├──────────────────────────────────────────►◄
      └─MIHTMOUT=units_of_time─┘
```

dependencies:	does not apply to MPC
range:	1–2147483647

Specifies the time (in tenths of a second) that allows you to adjust the missing interrupt handling interval during channel-to-channel communication between two VTAMs or between VTAM and an IBM 3172 Interconnect Controller.

Note: This start option does not apply to attachments between two hosts that use the multipath channel (MPC) feature. See “Multipath Channel Support” on page 66 and in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on MPC.

You can change the value of MIHTMOUT with the MODIFY VTAMOPTS command while VTAM is running.

MSGLEVEL

```
      ┌─MSGLEVEL=BASE──────────┐
►►────┤                        ├──────────────────────────────────────────►◄
      └─MSGLEVEL=──┬─BASE──┬──┘
                   ├─V4R1──┤
                   ├─V4R2──┤
                   ├─V4R3──┤
                   ├─V4R4──┤
                   └─CS390─┘
```

Specifies the version of the messages to use for those messages listed in *OS/390 SecureWay Communications Server: SNA Messages*. If the MSGLVL operand is specified on the USSMSG macroinstruction, it overrides the value specified on the MSGLEVEL start option.

Note: You can abbreviate the MSGLEVEL start option to MSGLVL.

You can change the value of MSGLEVEL with the MODIFY VTAMOPTS command while VTAM is running.

MSGLEVEL=BASE
Specifies that for all messages, except those that are coded otherwise in the USS table, VTAM issues the pre-V4R1 message that this message number replaces.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a new message or a message group.

If you specify MSGLEVEL=BASE and there are duplicate names in your SNA interconnected networks, you might not know which resource the message applies to because the message does not use a network-qualified name.

MSGLEVEL=V4R1
Specifies that VTAM issues the V4R1 message or messages, except for those that are coded otherwise in the USS table.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a V4R1 message or message group.

MSGLEVEL=V4R2
Specifies that VTAM issues the V4R2 message or messages, except for those that are coded otherwise in the USS table.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a V4R2 message or message group.

MSGLEVEL=V4R3
Specifies that VTAM issues the V4R3 message or messages, except for those that are coded otherwise in the USS table.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a V4R3 message or message group.

MSGLEVEL=V4R4
Specifies that VTAM issues the V4R4 message or messages, except for those that are coded otherwise in the USS table.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a V4R4 message or message group.

MSGLEVEL=CS390
Specifies that VTAM issues the CS for OS/390 message or messages, except for those that are coded otherwise in the USS table.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages replaced by a CS for OS/390 message or message group.

MSGMOD

```
┌─MSGMOD=NO──────┐
►►──┤                ├──────────────────────────────────────►◄
└─MSGMOD=──┬─NO──┬─┘
           └─YES─┘
```

Specifies whether VTAM should identify the issuing VTAM module in a message.

You can change the value of MSGMOD with the MODIFY VTAMOPTS command or the MODIFY MSGMOD command while VTAM is running.

MSGMOD=NO
Specifies that VTAM does not identify the issuing VTAM module in a VTAM message.

MSGMOD=YES
Specifies that VTAM inserts the last five characters of the name of the VTAM module that issued a VTAM message into each VTAM message following the message identifier.

Note: Messages might be truncated if you specify this option.

MXSAWBUF

```
►►─┬─MXSAWBUF=10000──────────────┬──────────────────────────────────────►◄
   │                             │
   └─MXSAWBUF=number_of_buffers──┘
```

range: 1–2147483647

Specifies the maximum number of buffers used for session awareness (SAW).

MXSSCPRU

```
►►─┬─MXSSCPRU=4096────────┬──────────────────────────────────────────►◄
   │                      │
   └─MXSSCPRU=ru_length───┘
```

range: 1–65535 bytes

Specifies the maximum length for RUs in SSCP sessions (SSCP-PU, SSCP-LU, and SSCP-SSCP).

MXSUBNUM

```
►►─┬─MXSUBNUM=511──────────┬──────────────────────────────────────────►◄
   │                       │
   └─MXSUBNUM=─┬─255──────┬─
               ├─511──────┤
               ├─1023─────┤
               ├─2047─────┤
               ├─4095─────┤
               ├─8191─────┤
               ├─16383────┤
               ├─32767────┤
               └─65535────┘
```

Specifies the maximum subarea number supported by a network to which a host VTAM is interconnected. If the value coded is not one of the values listed above, VTAM will not accept this start option for processing.

Note: The value specified on the HOSTSA start option cannot exceed the value specified for MXSUBNUM.

NCPBUFSZ

```
     ┌─NCPBUFSZ=512──────┐
►►───┤                   ├──────────────────────────────────────────────►◄
     └─NCPBUFSZ=──┬─512──┬─┘
                  ├─1024─┤
                  └─2048─┘
```

dependencies: see notes

Specifies the request unit size in bytes (I/O buffer size minus the size of the SNA header) used by VTAM when loading or performing a static dump of a remote NCP.

Notes:

1. This value does not apply to a MODIFY LOAD command. The length of the IPLTEXT RU is always 2048 plus the length of the SNA header.
2. This value applies only to static dumps or loads of 3720, 3725, and 3745 communication controllers. VTAM ignores this value when loading or dumping a 3705, and uses a data size of 512 bytes.

NETID

```
►►──NETID=network_id──────────────────────────────────────────────────►◄
```

Specifies the required 1–8 character identifier of the network containing the host.

Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in some countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network names (NETIDs). Although VTAM allows you to use #, other products might enforce this restriction.

NETID should be unique to each network within a set of SNA interconnected networks. Within each network, each SSCP must have the same NETID. Besides NETID, a gateway configuration requires GWSSCP=YES and an SSCPNAME.

NMVTLOG

```
     ┌─NMVTLOG=NPDA──────┐
►►───┤                   ├──────────────────────────────────────────────►◄
     └─NMVTLOG=──┬─ALWAYS─┬─┘
                 ├─NEVER──┤
                 └─NPDA───┘
```

Specifies when VTAM is to record network management vector transport (NMVT) records in LOGREC.

NMVTLOG=ALWAYS
Specifies that VTAM always records NMVT alerts in LOGREC.

NMVTLOG=NEVER
Specifies that VTAM never records NMVT alerts in LOGREC.

NMVTLOG=NPDA
Specifies that VTAM records NMVT alerts in LOGREC only when the NetView hardware monitor, NPDA, is not active.

NNSPREF

```
►►─┬─NNSPREF=NONE──────────────────┬──────────────────────────────────────►◄
   └─NNSPREF=network_node_server───┘
```

Specifies whether or not network node server dynamic switching is enabled for this end node.

Note: NNSPREF is valid only if the NODETYPE=EN start option is also used.

NNSPREF=NONE
Specifies that dynamic switching of the network node server will not be enabled.

If NNSPREF=NONE is specified through MODIFY VTAMOPTS, dynamic switching will be disabled. Existing CP-CP sessions with the current NNS will not be affected.

NNSPREF=*network_node_server*
Specifies the end node's preferred network node server. The name specified can be fully qualified with a network ID. If the network node server specified is not qualified with a network ID, the network ID of the end node will be used as the default. By specifying this start option with the name of a preferred network node server, the EN will dynamically switch CP-CP sessions from its current network node server to the preferred network node server when a CP-capable physical link is activated with the preferred network node server.

In addition, when CP-capable links specified in the VTAM configuration list are automatically activated at VTAM startup, only the link to the preferred network node server (if one exists) will trigger CP-CP session activation. If CP-CP sessions to the preferred network node server have not been initiated by the end of VTAM initialization, the network node server (NETSRVR) list will then be used to select a network node server.

This start option is modifiable using the MODIFY VTAMOPTS command. The network node specified for NNSPREF on the MODIFY VTAMOPTS command does not have to be predefined in the active network node server list and there does not have to be a nameless entry to handle it in the absence of an explicit entry. The value specified on the MODIFY command effectively gets dynamically added to the active NETSRVR list. This entry becomes the preferred entry and will have a higher priority than any other entry in the current NETSRVR list. The switching to the new preferred network node server will take place automatically if there is already an active CP-capable physical link to the new server.

Note: The switching could cause new session requests for resources on the EN to fail until the switching is complete.

NODELST

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─NODELST=name─────┘
```

Provides the 3–8 character name of the configuration restart file in which VTAM maintains a list of all currently active major nodes.

NODETYPE

```
►►──┬──────────────────────────────────┬──────────────────────────►◄
    │                      (1)         │
    └─NODETYPE=──┬─EN─┬──────────────────┘
                 └─NN─┘
```

Notes:

1 NODETYPE enables APPN function. The combination of HOSTSA, NODETYPE, and SACONNS determines the configuration (subarea node, interchange node, migration data host, network node, or end node).

Possible node configurations and their functional ability are summarized in Table 65 on page 596.

Specifies the node type that is presented to other APPN nodes. When NODETYPE is not specified, the node operates as a subarea node and all type 2.1 connections are managed as LEN connections. Also, the new start options, definition statements, and operands pertinent to APPN function are ignored.

When NODETYPE is specified, the HOSTSA and/or SACONNS start options determine whether this VTAM will also function as a subarea node. If HOSTSA= is coded and SACONNS= is not coded or SACONNS=YES is coded, then this VTAM will provide subarea function. If HOSTSA= is not coded, or HOSTSA= is coded and SACONNS=NO is coded, then this VTAM will not provide subarea function.

If a pure APPN node is desired, then the HOSTSA start option should not be coded or SACONNS=NO must be coded. Additionally, the DYNASSCP, GWSSCP, and MAXSUBA start options should not be coded because they have no effect. If you code them, the node will come up, but you will get error messages.

NODETYPE=EN
Specifies that the node is an end node. When NODETYPE=EN and subarea function has been enabled (see above), the node operates as a migration data host. If subarea function has not been enabled, the node operates as a pure end node. When the node is configured as a migration data host or end node, attempts to activate an NCP will fail. NCPs can be contacted, however.

NODETYPE=NN
Specifies that the node is a network node. When NODETYPE=NN and subarea function has been enabled (see above), the node operates as an interchange node. If subarea function has not been enabled, the node operates as a pure network node. Interchange nodes can own NCPs and their resources.

NOTRACE

For a description of the NOTRACE start option, see “TRACE for MODULE, STATE (with OPTION), or VTAM Internal Trace” on page 642 and “TRACE for Buffer, I/O, NCP Line, SIT, SMS, or STATE (with ID) Traces” on page 638.

NQNMODE

```
►►─┬─NQNMODE=NAME──────────┬──────────────────────────────────────────►◄
   │                       │
   └─NQNMODE=──┬─NAME──┬───┘
               └─NQNAME─┘
```

Specifies whether VTAM defines cross-network resources by both their non-network-qualified names and their network-qualified names or only by their network-qualified names.

If NQNMODE is specified on the CDRSC definition statement, it overrides the value specified on the NQNMODE start option. Dynamic CDRSCs and independent LUs which are defined using LOCADDR=0 are defined with the NQNMODE start option.

You can change the value of NQNMODE with the MODIFY VTAMOPTS command while VTAM is running.

NQNMODE=NAME
Specifies that the cross-network resource is defined by both its network-qualified name and its non-network-qualified name. The same-network CDRSC is always considered to be defined using NQNMODE=NAME because it is always identified by its non-network-qualified name. This is the mode in which pre-V4R1 releases of VTAM operate.

If a CDRSC is defined using NQNMODE=NAME (either by definition or start option), then no other resource can be defined to that host by the same non-network-qualified name.

NQNMODE=NQNAME
Specifies that the cross-network resource is defined by its network-qualified name and is not defined to this host by its non-network-qualified name.

If you have duplicate names in your SNA interconnected networks and prefer not to use other forms of alias name translation, you can specify (either by definition or start option) NQNMODE=NQNAME.

NSRTSIZE

```
►►──┬──────────────────────────────────────────────────────────────────►◄
    │
    └─NSRTSIZE=(─netid,size──)─┘
```

range (size): 1–2097148

Specifies the number of queue pointers in the symbol resolution table (SRT) for the network identifier(s) you specify. VTAM uses the value when it allocates an SRT directory size for a specific network identifier. NSRTSIZE allows you to override OSRTSIZE for a maximum of nine network identifiers.

You can code a maximum of nine *netid,size* pairs. *netid* can be any network identifier other than the VTAM host network identifier. If you specify the host network identifier, it is ignored and *size* is used.

For network identifiers not in the list, the value specified or defaulted for OSRTSIZE is used. The OSRTSIZE start option defines the directory size for all network identifiers not defined by the NSRTSIZE option.

NUMTREES

```
     ┌─NUMTREES=100──────────────────────────┐
►►───┤                                       ├──────────────────────────────────────►◄
     │                                  (1)  │
     └─NUMTREES=number_of_routing_trees──────┘
```

Notes:

1 NUMTREES is meaningful only if the NODETYPE=NN start option is also used.

range: 2–10000

Specifies a limit on the number of routing trees retained in the topology and routing services tree cache. Topology and routing services uses the trees when it receives requests for session routes. When the limit is reached, the oldest unused routing trees are discarded.

Setting a high number for NUMTREES makes session setup faster, but takes up more storage than a low number. If needed, you can free some storage by setting a lower number for NUMTREES, but session setup will then take longer because routes will have to be calculated.

You can change the value of NUMTREES with the MODIFY VTAMOPTS command while VTAM is running.

OSIEVENT

```
     ┌─OSIEVENT=PATTERNS──────┐
►►───┤                        ├──────────────────────────────────────────────────►◄
     │                        │
     └─OSIEVENT=──┬─ALL──────┬─┘
                  ├─NONE─────┤
                  └─PATTERNS─┘
```

Specifies whether VTAM topology notifications are generated for CMIP events. CMIP services generates and processes a great number of status notifications from resources even though only a small percentage of the notifications are needed by a manager application program.

The OSIEVENT start option lets you adjust the number of events reported, according to the requirements of the manager application programs. For details on how you specify the requirements of manager application programs and how they send requests to CMIP services, refer to the *OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide*.

You can change the value of OSIEVENT with the MODIFY VTAMOPTS command while VTAM is running. If you have coded OSIMGMT=YES, you can specify OSIEVENT=NONE or OSIEVENT=ALL with the command. If you have coded OSIMGMT=NO, you can specify any value.

OSIEVENT=ALL
Specifies that the VTAM topology agent generates notifications for all CMIP events. Specify OSIEVENT=ALL only if this VTAM is managed by a CMIP application program that uses filters to collect VTAM topology information but the filters do not match the patterns recognized by CMIP services. These patterns are described under in *OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide*.

With OSIEVENT=ALL, VTAM processes all event reports from network resources in that VTAM, regardless of the reporting flags set by the manager application program for that VTAM. The manager application program sees only the event reports matching the filters it created.

Be aware that specifying OSIEVENT=ALL can affect VTAM performance. However, the OSIEVENT=ALL start option allows the use of filters that do not match the patterns recognized by CMIP services.

OSIEVENT=NONE
Specifies that the VTAM topology agent creates no notifications for CMIP events, regardless of the reporting flags set by the manager application program for that VTAM.

OSIEVENT=PATTERNS
Specifies that the VTAM topology agent creates notifications only if there are active filters that CMIP services recognizes as requesting information for specific resources or classes of resources.

OSIEVENT=PATTERNS allows the use of certain filters that notify VTAM which specific resources, or classes of resources, should be monitored for event reports in that VTAM.

With this value, you can limit CPU cycles and storage use to the minimum needed to process the event reports from the selected resources.

OSIMGMT

```
►►─┬─OSIMGMT=NO──────────┬──────────────────────────────────────────►◄
   │                     │
   └─OSIMGMT=─┬─YES─┬────┘
              └─NO──┘
```

Specifies whether the VTAM topology agent code and the CMIP services code is to be loaded and activated for use.

You can change the value of OSIMGMT with the MODIFY VTAMOPTS command while VTAM is running.

OSIMGMT=YES
Specifies that the VTAM topology agent code and the CMIP services code is to

be loaded and activated for use. If you want to support CMIP application programs on VTAM, including the VTAM topology agent, specify YES for this start option.

If you specify OSIMGMT=YES, a directory definition file is required. See “CMIP Services Directory Definition File” on page 681 for information on coding the directory definition file. For further information on the CMIP services directory definition file, refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

OSIMGMT=NO
Specifies that the VTAM topology agent code and the CMIP services code is not to be loaded. If you do not want to use the VTAM topology agent or CMIP services on this host, specify NO.

OSITOPO

```
►►─┬─OSITOPO=(ILUCDRSC)──────────┬──────────────────────────────────────►◄
   │                             │
   └─OSITOPO=─┬─ALLCDRSC─┬────────┘
              └─ILUCDRSC─┘
```

Specifies whether the VTAM topology agent reports logical lines and all types of cross domain resources (CDRSCs) to CMIP manager application programs.

You can change the value of OSITOPO with the MODIFY VTAMOPTS command while VTAM is running, with the following restriction. When the current value of OSIMGMT=NO, you can specify any value for OSITOPO on the command. But, when the current value for OSIMGMT=YES, you can specify only OSITOPO=ILUCDRSC.

OSITOPO=ALLCDRSC
Specifies that the VTAM topology agent reports all CDRSCs to the CMIP manager application program as part of the local topology data. This value includes:
- Independent LUs (dynamic or predefined)
- Predefined CDRSCs
- Dynamic CDRSCs (except dynamic aliases)

OSITOPO=ILUCDRSC
Specifies that the VTAM topology agent reports all independent LUs (dynamic or predefined), but does not report other types of CDRSCs.

OSRTSIZE

```
►►─┬─OSRTSIZE=43──────────────────────────┬──────────────────────────────────►◄
   │                                      │
   └─OSRTSIZE=number_of_queue_pointers────┘
```

range: 1–2097148

Specifies the number of queue pointers in the symbol resolution table (SRT) directory for networks other than the VTAM host node’s network.

PDTRCBUF

```
   ┌─PDTRCBUF=2──────────────────┐
►►─┤                             ├──────────────────────────────────────►◄
   └─PDTRCBUF=number_of_buffers──┘
```

range: 2–255

Specifies the number of buffers to build for a session monitor request.

You can change the value of PDTRCBUF with the MODIFY VTAMOPTS command while VTAM is running.

PIUMAXDS

```
   ┌─PIUMAXDS=200────────────────┐
►►─┤                             ├──────────────────────────────────────►◄
   └─PIUMAXDS=calculation_factor─┘
```

range: 0–2147483647

A factor used in an equation to calculate the maximum buffer limit for PIU data space.

You can change the value of PIUMAXDS with the MODIFY VTAMOPTS command while VTAM is running.

PLUALMSG

```
   ┌─PLUALMSG=NOSUPP─────────┐
►►─┤                         ├──────────────────────────────────────►◄
   └─PLUALMSG=──┬─SUPPRESS─┬─┘
                └─NOSUPP───┘
```

Specifies whether VTAM is to suppress duplicate session setup failure messages associated with LOGAPPL failures to the same PLU.

You can change the value of PLUALMSG with the MODIFY VTAMOPTS command while VTAM is running.

PLUALMSG=NOSUPP
Specifies that VTAM does not suppress the session setup failure messages.

PLUALMSG=SUPPRESS
Specifies that VTAM suppresses duplicate session setup failure messages.

PPOLOG

```
   ┌─PPOLOG=NO──────┐
►►─┤                ├──────────────────────────────────────────────────►◄
   └─PPOLOG=──┬─NO──┬┘
              └─YES─┘
```

Specifies how much information is to be recorded in the primary program operator (PPO) log. When a PPO is running, the following events are always recorded in the PPO log:

- Commands issued by the program operator
- Messages issued by VTAM in response to those commands
- Unsolicited VTAM messages (for example, messages pertaining to alerts)

Note: When the NetView program is running, it is always the PPO.

The PPOLOG start option allows you to record the following additional events in the PPO log:

- Commands entered at the system console
- Messages issued by VTAM in response to those commands.

PPOLOG is supported by the NetView program. It should be used only if the PPO can support the option.

You can change the value of PPOLOG with the MODIFY VTAMOPTS command or the MODIFY PPOLOG command while VTAM is running.

PPOLOG=NO
Specifies that VTAM commands entered at the system console and messages VTAM issues in response are not recorded in the PPO log.

PPOLOG=YES
Specifies that all VTAM commands entered at the system console (except START and HALT) and all messages VTAM issues in response are recorded in the PPO log.

Note: Messages that are suppressed by the MODIFY SUPP command or by the SUPP start option are recorded in the PPO log, but do not appear on the system console. However, if multiple console support (MCS) is included in the system, all suppressed messages, originally destined for an MVS console, are sent to the hardcopy log.

PROMPT and NOPROMPT

```
            (1)
   ┌─PROMPT─────┐
►►─┤            ├──────────────────────────────────────────────────►◄
   └─NOPROMPT───┘
```

Notes:

1 A VTAM operator cannot enter the PROMPT or NOPROMPT start option; it can be coded only in ATCSTR00. The value coded in ATCSTR00 is ignored if start options are entered on the START command or if VTAM finds an error

in a start list. Upon finding an error in a start list, VTAM prompts the operator so that the operator can specify the option correctly.

Specifies whether VTAM prompts the VTAM operator to enter VTAM start options. PROMPT and NOPROMPT can be coded only in the ATCSTR00 predefined list of start options. You cannot code PROMPT in any other predefined start option list. If you include PROMPT or NOPROMPT in a predefined list other than ATCSTR00, the option will be ignored.

If you code NOPROMPT in ATCSTR00, VTAM does not prompt the operator. If NOPROMPT appears in ATCSTR00, you cannot override it by including PROMPT on the START command or in a predefined list. Do not put NOPROMPT in ATCSTR00 unless you are sure that a VTAM operator will never want to change the VTAM start options.

Note: If start options are entered with the START command, PROMPT is ignored.

PSRETRY

```
►►─┬─PSRETRY=(0,0,0,0)──────────────────────────────────────┬──────────────►◄
   │                                                        │
   └─PSRETRY=(low_tp_int,med_tp_int,high_tp_int,network_tp_int)─┘
```

range: 1 min–24 hours (expressed as nS, nM, or nH)

Specifies the interval between attempts to switch a rapid transport protocol (RTP) session to another path which may be equal to or better than the current path. If you do not specify an S, M, or H, seconds are assumed.

You can specify different time limits for low, medium, high, and network transmission priority connections. For example, you might want a batch connection to wait longer intervals than an interactive connection before looking for another path which may be equal to or better than the current path.

Specifying too short an interval might have a negative effect on network performance, especially if there are a large number of RTP connections.

You can change the value of PSRETRY with the MODIFY VTAMOPTS command while VTAM is running.

PSSTRACE

```
►►─┬─PSSTRACE=NORB──────┬──────────────────────────────────────────────►◄
   │                    │
   └─PSSTRACE=─┬─NORB─┬─┘
               ├─IRB──┤
               ├─SRB──┤
               └─BOTH─┘
```

Specifies whether IRB trace entries, SRB trace entries, both, or neither should be generated when the VTAM internal trace (VIT) is turned on with OPTION=PSS or OPTION=ALL.

PSSTRACE=IRB
Includes IRB entries in the VTAM internal trace.

PSSTRACE=SRB
Includes SRB entries in the VTAM internal trace.

PSSTRACE=BOTH
Includes IRB and SRB entries in the VTAM internal trace.

PSSTRACE=NORB
Does not include IRB nor SRB entries in the VTAM internal trace.

PSWEIGHT

```
   ┌─PSWEIGHT=LESSTHAN──────┐
►►─┤                        ├──────────────────────────────────────────►◄
   └─PSWEIGHT=──┬─EQUAL────┬┘
                ├─LESSTHAN─┤
                └─SAMEROUT─┘
```

Specifies the weight comparison between an old route and a newly calculated route before a path switch is attempted.

PSWEIGHT=EQUAL
Indicated that a path switch is desired between equal weight routes.

PSWEIGHT=LESSTHAN
Indicates that a path switch is desired only if the newly calculated route is less than the current weight.

PSWEIGHT=SAMEROUT
Indicates that PSRETRY will attempt to switch a rapid transport protocol (RTP) session even if the path switch results in the selection of the same route. This option will allow for a route to be switched to a more desirable VR within a VR-TG with multiple VRs.

RESUSAGE

```
                 (1)
   ┌─RESUSAGE=100──────────┐
►►─┤                       ├──────────────────────────────────────────►◄
   └─RESUSAGE=usage_limit─┘
```

Notes:

1 RESUSAGE is meaningful only if the NODETYPE=NN start option is also used.

range: 0–2147483647

Specifies how many times a resource (node or TG) is used during route selection before that resource is considered overused. After a resource is overused, any routing trees using it will be reconstructed in an attempt to use alternate resources. RESUSAGE is valid only when NODETYPE=NN.

Note: RESUSAGE=0 indicates that the resource is never considered to be overused.

You can change the value of RESUSAGE with the MODIFY VTAMOPTS command while VTAM is running.

ROUTERES

```
                    (1)
   ┌─ROUTERES=128──────────────┐
►►─┤                           ├──────────────────────────────────────────►◄
   └─ROUTERES=resistance_value─┘
```

Notes:

1 ROUTERES is meaningful only if the NODETYPE=NN start option is also used.

range: 0–255

Specifies a route addition resistance value. This is a relative measure of the desirability for this node to perform intermediate session routing function. The value specified for this node is compared to the values of other network nodes during route calculation. The lower the value, the more desirable it is to have this node provide intermediate session routing.

You can change the value of ROUTERES with the MODIFY VTAMOPTS command while VTAM is running.

SACONNS

```
   ┌─SACONNS=YES──────┐
►►─┤                  ├──────────────────────────────────────────────────►◄
   └─SACONNS=──┬─NO──┬─┘
               └─YES─┘
```

Specifies if subarea connections will be used. SACONNS need only be specified when defining a pure APPN EN or NN that will use a HOSTSA number other than the default value (1). For details on defining various node configurations, see Table 65 on page 596.

SACONNS=NO
When NODETYPE is coded, indicates that this VTAM will not provide subarea function. (That is, this VTAM will operate as a pure APPN node.)

SACONNS=YES
Indicates that this VTAM will provide subarea function.

SAVERSCV

```
                        (1)
     ┌─SAVERSCV=(NO,KEEP)──────────┐
►►───┼─────────────────────────────┼──────────────────────────────────────►◄
     └─SAVERSCV=──┬─(,FREE)──────┬─┘
                  ├─(,KEEP)──────┤
                  ├─NO───────────┤
                  ├─(NO,FREE)────┤
                  ├─(NO,KEEP)────┤
                  ├─YES──────────┤
                  ├─(YES,FREE)───┤
                  ├─(YES,KEEP)───┤
                  ├─SAWDATA──────┤
                  ├─(SAWDATA,FREE)┤
                  └─(SAWDATA,KEEP)┘
```

Notes:

1 SAVERSCV is meaningful only if NODETYPE is also used.

Specifies whether or not Route Selection Control Vectors (RSCVs) for future sessions are to be saved for display purposes following the completion of session activation. SAVERSCV also specifies whether RSCV storage for existing sessions should be freed immediately. By default, these RSCVs are not saved. If YES or SAWDATA is specified, the D NET, SESSIONS, PATHINFO=YES, SID= command can then be used to display the saved RSCVs for the particular session specified.

You can change the value of SAVERSCV with the MODIFY VTAMOPTS command while VTAM is running.

Note: The secondary values, FREE and KEEP, have no effect at VTAM START time.

SAVERSCV=(,FREE)
Specifies that RSCV storage for existing sessions is to be freed immediately. The primary value currently in effect (YES,NO, or SAWDATA) remains unchanged.

SAVERSCV=NO
Specifies that RSCVs for future sessions are not saved following the completion of session activation. By default, RSCVs for existing sessions will remain.

SAVERSCV=(NO,FREE)
Specifies that RSCV storage for future sessions are to be freed following session activation, and that RSCV storage for existing sessions is to be freed immediately.

SAVERSCV=(NO,KEEP)
Specifies that RSCV storage for future sessions are to be freed following session activation, but that RSCV storage for existing sessions will remain.

SAVERSCV=YES
Specifies that RSCVs for future sessions are to be saved following the completion of session activation.

SAVERSCV=SAWDATA
Specifies that only the RSCVs associated with any sessions matching the PLU/SLU combination in the SAW data filter table are saved. This option can be used whether or not NetView is running on the given VTAM. If a new SAW data filter table is activated and if SAVERSCV=SAWDATA, the SAW data filter table

will control which RSCVs are reported to NLDM and will impact the saving and reporting of RSCVs associated with future sessions. The reporting of RSCVs for sessions that are already active will be governed by the previous SAW data filter table.

Note: Changing the SAVERSCV option from YES to NO, or from SAWDATA to NO, is intended to indicate that RSCV storage for future sessions is not to be saved.

SAWMAXDS

```
►►─┬─SAWMAXDS=100──────────────────┬──────────────────────────────────────────►◄
   └─SAWMAXDS=calculation_factor───┘
```

range: 1–2147483647

A factor used in an equation to calculate the maximum buffer limit for SAW data space.

SAWMXQPK

```
►►─┬─SAWMXQPK=0──────────────────┬──────────────────────────────────────────►◄
   └─SAWMXQPK=calculation_factor─┘
```

range: 0–2147483647

Used to calculate the number of session awareness (SAW) data space buffers to queue before packing SAW buffers.

You can change the value of SAWMXQPK with the MODIFY VTAMOPTS command while VTAM is running.

SDLCMDRS

```
►►─┬─SDLCMDRS=(STATS,INOPS)──────────────────────────┬──────────────────────────►◄
   ├─SDLCMDRS=YES────────────────────────────────────┤
   │                                                 │
   └─SDLCMDRS=──┬─YES──────────────────────────────┬─┘
                ├─NO───────────────────────────────┤
                ├─STATS────────────────────────────┤
                ├─NOSTATS──────────────────────────┤
                ├─(,INOPS)─────────────────────────┤
                ├─(,NOINOPS)───────────────────────┤
                └─(──┬─STATS──┬─,INOPS──────┬──────┬─)─┘
                     │        └─,NOINOPS────┘      │
                     └─NOSTATS──┬─,INOPS──────┬────┘
                                └─,NOINOPS────┘
```

Specifies whether VTAM records SDLC inoperative and statistical miscellaneous data records (MDRs) in LOGREC.

Note: If you do not record MDRs, and a problem occurs in an SDLC device, problem diagnosis can be difficult because there are no MDRs in LOGREC.

STATS,NOSTATS and YES,NO mean the same. STATS,NOSTATS is the preferred coding. YES,NO is supported on this start option for compatibility with releases prior to VTAM V3R4.1.

You can change the value of SDLCMDRS with the MODIFY VTAMOPTS command while VTAM is running.

SDLCMDRS=YES
Specifies that VTAM records the SDLC statistical MDRs in LOGREC.

SDLCMDRS=STATS
Specifies that VTAM records the SDLC statistical MDRs in LOGREC.

SDLCMDRS=NO
Specifies that VTAM does not record the SDLC statistical MDRs in LOGREC.

SDLCMDRS=NOSTATS
Specifies that VTAM does not record the SDLC statistical MDRs in LOGREC.

SDLCMDRS=INOPS
Specifies that VTAM records the SDLC inoperative MDRs in LOGREC.

SDLCMDRS=NOINOPS
Specifies that VTAM does not record SDLC inoperative MDRs in LOGREC.

SECLVLCP

```
                    (1)
  ┌─SECLVLCP=ADAPT──────┐
►►┼──────────────────────┼──────────────────────────────────────────────►◄
  └─SECLVLCP=──┬─LEVEL1─┬┘
               ├─LEVEL2─┤
               └─ADAPT──┘
```

Notes:

1 The SECLVLCP start option is meaningful only if the NODETYPE and VERIFYCP start options are also used.

dependency: LU 6.2 sessions

Specifies whether VTAM uses the basic or enhanced protocol for session-level security verification during activation of an LU 6.2 session involving control points.

Some VTAMs allow only basic protocol, while others allow basic or enhanced. VTAMs that allow both protocols fall into two classes:

- Those that allow choice of basic or enhanced protocol at any time
- Those that allow a restricted choice: after the enhanced protocol is used to a particular partner, only the enhanced protocol can be subsequently used to that partner.

SECLVLCP=LEVEL1
Specifies that the basic protocol is used. If a partner CP does not support the basic protocol, VTAM rejects the session and issues a sense code.

LEVEL1 is useful in a VTAM that is connected to an earlier version of VTAM that is using session-level security verification. LEVEL1 prevents the restricted-choice class of VTAMs from locking out subsequent sessions to the earlier version of VTAM.

SECLVLCP=LEVEL2
Specifies that the enhanced protocol is used. If a partner CP does not support the enhanced protocol, VTAM rejects the session and issues a sense code. LEVEL2 is useful when all communicating VTAMs have session-level security, and you want to prevent sessions being set up with the less stringent LEVEL1 protocol.

SECLVLCP=ADAPT
Specifies that VTAM will negotiate whether to use the basic or enhanced protocol for LU-LU session-level security verification, depending on the protocol supported by the partner CP. VTAM attempts to use the enhanced protocol but will use the basic protocol if a partner CP does not support the enhanced protocol.

ADAPT is useful in a VTAM that communicates with those VTAMs using LEVEL1 and with the restricted-choice class of VTAMs.

SIRFMSG

```
►►─┬─SIRFMSG=ALLSSCP──────────┬──────────────────────────────────────►◄
   │                          │
   └─SIRFMSG=──┬─ALLSSCP─┬────┘
               ├─OLUSSCP─┤
               └─NONE────┘
```

Controls the display of IST663I, IST664I, IST889I, and their message groups. These messages are issued when a session initiation request fails.

You can change the value of SIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

SIRFMSG=ALLSSCP
Specifies that messages are issued in all SSCPs.

SIRFMSG=OLUSSCP
Specifies that messages are issued only in the origin logical unit (OLU) SSCP.

SIRFMSG=NONE
Specifies that no messages are issued in any SSCP.

SLOWVAL

```
►►─┬─SLOWVAL=(0,0)──────────────────────┬──────────────────────────────►◄
   │                                    │
   └─SLOWVAL=──┬─warn_time──────────────┬┘
               └─(warn_time,end_time)───┘
```

range: 5-999

Specifies that a network node which has been placed in a SLOWDOWN mode can now allow a maximum SLOWDOWN time to be set.

SLOWVAL=warn_time
Specifies the amount of time a resource can be in SLOWDOWN before a ″warning message″ will be sent to the console stating that the resource has been in slowdown for set amount of time. The device can be automotically removed from SLOWDOWN after the SLOWVAL=*end_time* is specified, or specify SLOWVAL=0 for an informational ″warning message″ whenever a device is in SLOWDOWN for a long period of time or if the device comes out of SLOWDOWN mode.

SLOWVAL=end_time
Specifies the amount of time a resource can be in SLOWDOWN mode before it is automatically taken out of SLOWDOWN mode by VTAM. After a ″warning message″ has been issued, a new message will be displayed as soon as that resource is taken out of SLOWDOWN mode (whether VTAM-initiated or DEVICE- initiated).

SLUALMSG

```
►►─┬─SLUALMSG=NOSUPP──────────┬──────────────────────────────────────►◄
   │                          │
   └─SLUALMSG=──┬─SUPPRESS─┬──┘
                └─NOSUPP───┘
```

Specifies whether VTAM is to suppress messages for a session setup failure caused when the SLU is not enabled for a session.

You can change the value of SLUALMSG with the MODIFY VTAMOPTS command while VTAM is running.

SLUALMSG=NOSUPP
Specifies that VTAM does not suppress the session setup failure messages.

SLUALMSG=SUPPRESS
Specifies that VTAM suppresses the session setup failure messages.

SMEAUTH

```
►►─┬─SMEAUTH=DISCARD──────────┬──────────────────────────────────────►◄
   │                          │
   └─SMEAUTH=──┬─DISCARD─┬────┘
               ├─EXIT────┤
               └─KEEP────┘
```

Controls session management exit (SME) authorization data without having to modify or code an SME. See *OS/390 SecureWay Communications Server: SNA Customization* for more information about the SME authorization data.

SMEAUTH=DISCARD
Prevents SME authorization data from being saved.

SMEAUTH=EXIT
Allows the SME to determine if the authorization data should be kept or freed when the session is activated.

SMEAUTH=KEEP
Saves the authorization data for all sessions.

SNAPREQ

```
   ┌─SNAPREQ=1000──────────────┐
►►─┤                           ├──────────────────────────────────────────►◄
   └─SNAPREQ=number_of_requests─┘
```

range: 1–32767

Specifies, when the SMS (buffer use) trace is active, the threshold number of requests for VTAM buffers after which to take a snapshot dump.

SNVC

```
            (1)
   ┌─SNVC=3──────────────────┐
►►─┤                         ├──────────────────────────────────────────►◄
   └─SNVC=subnet_visit_count─┘
```

Notes:

1 SNVC is meaningful only if the BN=YES start option is also used.

range: 1–255

Defines a maximum number of networks this border node will search when looking for a resource. This start option is valid only when BN=YES for this node.

You can change the value of SNVC with the MODIFY VTAMOPTS command while VTAM is running.

Notes:

1. If SNVC=1 is coded, the search scope is restricted to the originating subnetwork. A value of SNVC=2 is required to allow a request to traverse one subnetwork boundary.
2. The maximum subnet count can be modified downwards if a node along the path has a lower SNVC count. However, the received SNVC is never modified upward.

SONLIM

```
   ┌─SONLIM=(60,30)──────────────────────┐
►►─┤                                     ├──────────────────────────────►◄
   └─SONLIM=──┬─(percent_1,percent_2)──┬─┘
              ├─(percent_1)────────────┤
              └─(,percent_2)───────────┘
```

range: 1–99

Defines the maximum percentage of fixed I/O buffers available for the session outage notification (SON) request/response units.

VTAM calculates the *percent_1* number of buffers and the *percent_2* value, using the values you specify as the percentage of the I/O buffers available for SON.

These numbers are calculated when VTAM is started, and are not affected by any increase in the number of I/O buffers due to dynamic buffer expansion.

The session outage notification does not pre-allocate buffers. No I/O buffers are allocated until SON processing begins.

SONLIM=percent_1
Specifies the percentage of the fixed I/O buffer pool that can be allocated for SON request units. VTAM calculates this percentage by using the *percent_1* value and the *baseno* value coded for IOBUF. This is the maximum number of buffers that can be allocated for notification request units. If VTAM cannot initiate all the SON request units (one buffer is required for each request unit) within the available storage, it suspends the notification process until the number of buffers allocated for SON request units drops below the calculated threshold. Then the process begins again and more SON request units are generated.

SONLIM=percent_2
Specifies the percentage of fixed I/O buffers that are still allocated to session outage notification request units when the notification process begins again. After the number of buffers allocated to session outage notification drops below the *percent_2* value, the process begins generating SON reports again.

Note: The value of *percent_2* must be less than the value of *percent_1*.

SORDER

```
                  (1)
  ┌─SORDER=APPN──────────┐
►►┼──────────────────────┼──────────────────────────────────────────────►◄
  └─SORDER=─┬─ADJSSCP──┬─┘
            ├─APPN──────┤
            ├─APPNFRST──┤
            └─SUBAREA───┘
```

Notes:

1 SORDER is meaningful only in an interchange node or a migration data host.

Controls the order in which the APPN and subarea networks are searched when a network search request is received from the subarea network. This search order may be overridden if the target LU's location is already identified and SSCPORD=PRIORITY is specified. SORDER is valid only for network nodes (NODETYPE=NN) functioning as interchange nodes, and end nodes functioning as migration data hosts. For further information on the interaction of the SORDER start option with other start options, see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

You can change the value of SORDER with the MODIFY VTAMOPTS command while VTAM is running.

For more granular control over the search order, a different SORDER can be specified for each individual adjacent SSCP table. Refer to the ″Adjacent SSCP table section″ of this book.

See “cdrmname” on page 505 for related information on network searches.

SORDER=ADJSSCP
Specifies that the search should be conducted based on the order coded in the adjacent SSCP table. If you specify SORDER=ADJSSCP, the APPN network is not searched unless you also coded the operand ISTAPNCP as one of the entries within the adjacent SSCP table.

SORDER=APPN
Specifies that searching the APPN network is preferred, regardless of whether (or where) the operand ISTAPNCP is coded in the adjacent SSCP table. However, the owning CDRM, the predefined owning CDRM, and any previously successful adjacent CDRMs (if SSCPORD=PRIORITY is specified), if known, are searched before the APPN network.

SORDER=APPNFRST
Specifies that the APPN network is to be searched first, without regard to whether (or where) the operand ISTAPNCP is coded in the adjacent SSCP table, or whether the owning CDRM, the predefined owning CDRM, or any previously successful adjacent CDRMs are known.

Note: SORDER=APPNFRST and SSEARCH=APPNFRST are only meaningful on Network Nodes. If SSEARCH=APPNFRST is coded on an EN, it will prevent finding resources on the EN's domain.

SORDER=SUBAREA
Specifies that the subarea network is to be searched first, regardless of whether (or where) the operand ISTAPNCP is coded in the adjacent SSCP table.

SRCHRED

```
   ┌─SRCHRED=OFF──────┐
►►─┤                  ├──────────────────────────────────────────────────►◄
   └─SRCHRED=──┬─OFF─┬─┘
               └─ON──┘
```

Specifies whether this node can reduce searches for resources which are found to be unreachable. Refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about using the search reduction facility.

You can change the SRCHRED value with the MODIFY VTAMOPTS command while VTAM is running.

SRCHRED=OFF
Specifies that this node does not reduce searches.

SRCHRED=ON
Specifies that this node does reduce searches.

SRCOUNT

```
                (1)
   ┌─SRCOUNT=10──────────────────────┐
►►─┤                                 ├──────────────────────────────────────►◄
   └─SRCOUNT=number_of_search_requests─┘
```

Notes:

1 SRCOUNT is meaningful only if the SRCHRED=ON start option is also used.

range: 0–65535

Specifies how many search requests can be limited before VTAM performs another resource discovery search. SRCOUNT is meaningful only when search reduction is active (SRCHRED=ON).

If SRCOUNT=0 is coded, then the search count threshold is not used.

You can change the value of SRCOUNT with the MODIFY VTAMOPTS command while VTAM is running.

SRTIMER

```
                (1)
   ┌─SRTIMER=30──────────────┐
►►─┤                         ├──────────────────────────────────────────────►◄
   └─SRTIMER=number_of_seconds─┘
```

Notes:

1 SRTIMER is meaningful only if the SRCHRED=ON start option is also used.

range: 0–65535 seconds

Specifies the number of seconds during which VTAM does not conduct searches for an unreachable resource. Once the time limit has expired, VTAM will conduct a search. SRTIMER is meaningful only when search reduction is active (SRCHRED=ON).

If SRTIMER=0 is coded, then the search time threshold is not used.

You can change the value of SRTIMER with the MODIFY VTAMOPTS command while VTAM is running.

SSCPDYN

```
                (1)
   ┌─SSCPDYN=YES─────────┐
►►─┤                     ├──────────────────────────────────────────────────►◄
   └─SSCPDYN=──┬─NO──┬──┘
               └─YES─┘
```

Notes:

1 The SSCPDYN start option applies only for interconnected networks (that is, GWSSCP=YES is used).

Specifies whether VTAM dynamically adds entries to the adjacent SSCP table.

If you specify SSCPDYN=NO, you might have to define additional adjacent SSCP tables. When an application issues a CLSDST PASS macroinstruction during

session setup, VTAM does not build an adjacent SSCP table in the application's network. (When SSCPDYN=YES is coded in the application's network, VTAM builds such a table automatically.)

This start option is closely related to SSCPORD. Refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how to use these two start options.

SSCPDYN=NO
Specifies that VTAM does not add new entries unless it is the owner of the resource. The owner is determined by the CDRM operand of the CDRSC definition statement, or by session initiation.

SSCPDYN=YES
Specifies that VTAM adds a new entry to a cross-domain resource's adjacent SSCP table whenever it receives a session initiation request from the resource through an SSCP that is not already in the table.

SSCPID

```
►►──SSCPID=sscp_id──────────────────────────────────────────────────────►◄
```

range: 0–65535

Specifies that part of an SSCP identifier used when a physical unit or external CDRM establishes contact with VTAM. This start option is required.

It is recommended that the SSCPID in each network and each connecting network be unique. If the SSCPs within a network do not have cross-network sessions, the SSCPIDs are not required to be unique. If the SSCPs do have cross-network sessions, the SSCP for each end of the cross-network sessions must have a different SSCPID. If more than one SSCP in one network has the same SSCPID, they cannot all have a session with the same SSCP in another network.

Notes:

1. VTAM determines the ACTCDRM contention winner by comparing the SSCPID of each SSCP. If ACTCDRM contention occurs on two SSCPs with the same SSCPID, the contention must be resolved by the network operators.
2. If you plan to incorporate a non-VTAM host, VTAM's SSCPID and HOSTSA start options values should be coded so that *both* values are either higher or lower than the non-VTAM host.

SSCPNAME

```
►►──SSCPNAME=name───────────────────────────────────────────────────────►◄
```

dependencies: see notes

Specifies the required 1–8 character name of the VTAM SSCP or CP. You should use the same name here that you use on the host CDRM in multiple-domain networks. This start option is required.

When you code the SSCPNAME start option, consider the items listed below. The first three items are requirements. The fourth item is a suggestion for improving usability and easing network management.

- The SSCPNAME must be the same as the name on the NAME operand of the corresponding GWNAU definition statement, if one is coded. (The GWNAU definition statement is used in the gateway NCP generation to predefine the gateway SSCP's alias network address within the gateway NCP.)

  For more information on the GWNAU definition statement, see the *NCP, SSP, and EP Resource Definition Reference*.
- The SSCPNAMEs and CDRM names must be unique among SSCPs that will participate in cross-domain or cross-network sessions.
- When VTAM is defined as a type 2.1 node to another VTAM or NCP on a switched SDLC connection, specify the SSCPNAME value on the CPNAME operand of the PU statement. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about casual connection.
- The SSCPNAME should match the name on the host CDRM definition statement in a subarea environment, and must match the host CDRM definition statement in an APPN or mixed APPN-subarea environment to avoid errors. See "cdrmname" on page 121 for further details.

SSCPORD

```
►►─┬─SSCPORD=PRIORITY──────────┬──────────────────────────────────────►◄
   │                           │
   └─SSCPORD=──┬─DEFINED─┬─────┘
               └─PRIORITY─┘
```

Specifies whether VTAM searches an adjacent SSCP table in priority order (the default) or in the order in which the table is defined.

You can change the value of SSCPORD with the MODIFY VTAMOPTS command while VTAM is running.

This start option is closely related to SSCPDYN. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how to use these two start options.

SSCPORD=DEFINED
Specifies that VTAM scans the table in the same order that SSCPs are listed, if past session initiation attempts were successful, with the exception that the owner, if identified, is tried first.

SSCPORD=PRIORITY
Specifies that VTAM gives preference to SSCPs it uses. The search order is as follows:

1. The CDRM that owns the resource (if identified).
2. The SSCPs for which the most recent session-initiation attempt succeeded.
3. The SSCPs for which no session-initiation attempt has been made.
4. The SSCPs for which the last session-initiation attempt failed.

SSDTMOUT

```
   ┌─SSDTMOUT=30──────────────────┐
►►─┤                              ├──────────────────────────────────────────►◄
   └─SSDTMOUT=number_of_seconds───┘
```

range: 1–65535

Specifies the number of seconds that can pass without a new LU-LU session beginning before VTAM automatically disconnects the switched PU.

You can change the value of SSDTMOUT with the MODIFY VTAMOPTS command while VTAM is running.

SSEARCH

```
                 (1)
   ┌─SSEARCH=YES──────────────────┐
►►─┤                              ├──────────────────────────────────────────►◄
   └─SSEARCH=──┬─YES────────┬─────┘
               ├─NO─────────┤
               ├─CACHE──────┤
               └─APPNFRST───┘
```

Notes:

1 SSEARCH is meaningful only if the NODETYPE=NN start option is also used.

Controls whether the subarea network is searched when a network search request is received from the APPN network. For further information on the interaction of the SSEARCH start option with other start options, see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

You can change the value of SSEARCH with the MODIFY VTAMOPTS command while VTAM is running.

SSEARCH=CACHE
Specifies that the subarea network is not to be searched unless it is known that the target LU is located in the subarea network. This knowledge might exist as a result of system definition or as a result of an earlier successful search.

SSEARCH=APPNFRST
Specifies that if SSEARCH=APPNFRST is coded on an EN, it will prevent finding resources on the EN's domain. It is only meaningful on Network Nodes.

SSEARCH=NO
Specifies that the subarea network is not to be searched.

SSEARCH=YES
Specifies that the subarea network is to be searched.

STRGR

```
   ┌─STRGR=ISTGENERIC──────────┐
►►─┤                           ├──────────────────────────────────────────►◄
   └─STRGR=──┬─ISTGENERIC────┬─┘
             ├─NONE──────────┤
             └─structure_name┘
```

Allows you to rename the generic resources structure.

STRGR is valid only when the coupling facility function is supported. For VTAMs that are to share the same generic resource information, you must specify the same STRGR value for each VTAM.

STRGR=ISTGENERIC
Specifies the IBM-supplied generic resources structure name.

STRGR=NONE
Specifies that this VTAM should not attempt to connect to ISTGENERIC, even if it is defined.

STRGR=*structure_name*
Specifies a user-defined generic resources structure name. *structure_name* can have up to 16 characters. It must have a minimum of three characters, and these must be the characters IST. It can contain numeric characters, alphabetic characters, national characters ($,@,#), or an underscore(_).

If the structure name specified on STRGR is not defined at VTAM initialization, then no connection attempt is made and the generic resource function is disabled.

STRMNPS

```
   ┌─STRMNPS=ISTMNPS───────────┐
►►─┤                           ├──────────────────────────────────────────►◄
   └─STRMNPS=──┬─ISTMNPS───────┬─┘
               ├─NONE──────────┤
               └─structure_name┘
```

Specifies the name of the generic resources structure to be used for multinode persistent sessions.

STRMNPS=ISTMNPS
Specifies the default name of the MVS coupling facility structure to be used for multinode persistent sessions.

STRMNPS=NONE
Specifies that VTAM will not check for the existence of multinode persistent coupling facility structure. Coding STRMNPS=NONE disables the multinode persistent function for this VTAM, even if a structure has been defined.

STRMNPS=*structure_name*
Specifies a 1–16-character generic resources structure name. The name can contain numeric characters, alphabetic characters, national characters ($, A, #) or an underscore (_). The name must begin with IST as the first 3 characters.

Note: If you are using multiple coupling facility structures for multinode persistent session support, the number of characters used to specify the

name of the base structure (STRMNPS start option) must account for characters needed for unique names for all alternate structures. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on using alternate structures.

SUPP

```
►►─┬─SUPP=NOSUP──────┬──────────────────────────────────────────►◄
   │                 │
   └─SUPP=──┬─NOSUP─┬─┘
            ├─INFO──┤
            ├─WARN──┤
            ├─NORM──┤
            └─SER───┘
```

Specifies the highest class of VTAM messages for which VTAM suppresses message output to the VTAM operator console, and suppresses transmission to a program operator, if one exists. Messages that are generated in response to an operator request (such as the DISPLAY command) and messages that require a response (prompting messages) cannot be suppressed.

If multiple console support (MCS) is included in the system, all suppressed messages, originally destined for an MVS console, are sent to the hardcopy log. These suppressed messages are also copied to the PPO log if PPOLOG=YES was specified.

If the destination for a suppressed message is a program operator, the message is lost.

You can change the value of SUPP with the MODIFY VTAMOPTS command or the MODIFY SUPP command while VTAM is running.

VTAM suppresses messages within a multiple line write-to-operator (MLWTO) group based on the suppression level of the first message in the group. VTAM ignores the suppression levels for subsequent messages within the group.

Refer to *OS/390 SecureWay Communications Server: SNA Messages* for individual suppression levels.

SUPP=NOSUP
Specifies that all VTAM messages are printed at the console. (NOSUP means “no suppression”).

SUPP=INFO
Specifies that informational messages are suppressed. Informational messages are those that inform the operator that commands or procedures have been accepted for processing.

SUPP=WARN
Specifies that warning messages (as well as informational messages) are suppressed. Warning messages identify error conditions that do not cause commands to fail or be rejected. These messages inform the operator that there is a problem, such as an operand that is not valid or a minor node that cannot be activated, but that VTAM can continue to process other parts of the command or procedure.

SUPP=NORM
Specifies that normal completion messages (as well as informational and warning messages) are suppressed. Normal completion messages inform the operator that commands have completed processing successfully, a configuration has been activated successfully, or a procedure has been terminated.

SUPP=SER
Specifies that serious error messages (as well as informational, warning, and normal completion messages) are suppressed. Serious error messages identify error conditions that cause commands or procedures to fail. These messages tell the operator that a command must be reentered or a procedure reinitiated.

Error messages that identify an even more serious situation, such as the abnormal termination of a user task or of VTAM itself, cannot be suppressed.

SWNORDER

```
     ┌─SWNORDER=CPNAME──────┐
►►───┤                      ├──────────────────────────────────────────────►◄
     └─SWNORDER=──┬─CPNAME─┬─┘
                  └─STATNID─┘
```

Specifies the way VTAM locates a switched PU. If you do not code SWNORDER, VTAM searches for a PU by the CPNAME first. If VTAM does not find the PU by the CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

You can change the value of SWNORDER with the MODIFY VTAMOPTS command while VTAM is running. You can override this start otion for specific resources by coding the SWNORDER operand on the GROUP or LINE statements.

SWNORDER=CPNAME
Specifies that VTAM searches for a switched PU by the CPNAME first.

SWNORDER=STATNID
Specifies that VTAM searches for a switched PU by the station identifier first.

Note: SWNORDER controls only regular (non-DLUR) switched PUs. For DLUR-defined PUs, see the DLRORDER start option.

TCPNAME

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └─TCPNAME=tcp_job-name─┘
```

Specifies the job name of the TCP stack through which VTAM communicates with the IP network. The value specified for *tcp_job-name* is the 1–8 character job name of the required TCP stack.

If this option is not specified, the first appropriate TCP/IP stack will be selected. The TCP/IP stack must be at a level to support Enterprise Extender. However, if there are multiple TCP/IP stacks you may need to specify one, as there are products

which predefine the IP address of partner nodes, and these nodes require CS for OS/390 to send through a given address and therefore, a given TCP/IP stack.

Note: TCPNAME can be dynamically modified using MODIFY VTAMOPTS but the new value will not be used until all lines in the XCA major node used for Enterprise Extender are inactive. However, displays of VTAM start options will show the new value immediately. Any subsequent line activation from the Enterprise Extender XCA major node will make use of the new TCPNAME value.

TNSTAT

```
                 ┌NOTNSTAT──────────────────────────────┐
►►───────────────┼──────────────────────────────────────┼──────────────────────────────►◄
                 │          ┌,NOCNSL┐    ┌,TIME=60──────┐│
                 └TNSTAT────┼───────┼────┼──────────────┘│
                            └,CNSL──┘    └,TIME=minutes──┘
```

Specifies whether VTAM keeps tuning statistics. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for further information on tuning.

If you code TNSTAT at start time, you can change the value with MODIFY TNSTAT command or the MODIFY NOTNSTAT command while VTAM is running.

Note: If you do not code TNSTAT at start time, you cannot use the MODIFY TNSTAT command at a later time to request that tuning statistics be kept.

System management facilities (SMF) is required to record tuning statistics.

TNSTAT
Specifies that tuning statistics should be kept.

NOTNSTAT
Specifies that tuning statistics should not be kept.

The following operands are valid only if you code TNSTAT:

CNSL
Specifies that the tuning statistics records are written to the console and to the SMF data set.

NOCNSL
Specifies that the tuning statistics records are written only to the SMF data set.

TIME=*number_of_minutes*
Specifies the number of minutes that should elapse between records. One record is written every *number_of_minutes* for each channel-attached SNA controller in the network. A record is also written when a controller is deactivated or when a MODIFY TNSTAT command or MODIFY NOTNSTAT command is issued.

The value for TIME must be in the range 1–1440. If you do not code TIME, or if you code a value for TIME outside the acceptable range, TIME=60 is used.

TRACE for Buffer, I/O, NCP Line, SIT, SMS, or STATE (with ID) Traces

```
   ┌─NOTRACE,TYPE=BUF (1)──────────────────┐
►►─┤                                       ├──────────────────────────►
   └─TRACE,TYPE=BUF─┤ BUF Trace Operands (2) ├─┘

   ┌─NOTRACE,TYPE=IO (1)───────────────────┐
►──┤                                       ├──────────────────────────►
   └─TRACE,TYPE=IO─┤ IO Trace Operands (2) ├─┘

   ┌─NOTRACE,TYPE=LINE (1)─────────────────┐
►──┤                                       ├──────────────────────────►
   └─TRACE,TYPE=LINE─┤ LINE Trace Operands (2) ├─┘

   ┌─NOTRACE,TYPE=SIT (1)──────────────────┐
►──┤                                       ├──────────────────────────►
   └─TRACE,TYPE=SIT─┤ SIT Trace Operands (2) ├─┘

   ┌─NOTRACE,TYPE=SMS (1)──────────┐
►──┤                               ├──────────────────────────────────►
   └─TRACE,TYPE=SMS─,ID=VTAMBUF (2)─┘

   ┌─NOTRACE,TYPE=STATE (1)────────────────┐
►──┤                                       ├──────────────────────────►◄
   └─TRACE,TYPE=STATE─┤ STATE Trace Operands (2) ├─┘
```

BUF Trace Operands:

```
├──,ID=node_name──┬──────────┬──┬─,AMOUNT=PARTIAL──────┬──────────────►
                  └─,EVERY───┘  │                      │
                                └─,AMOUNT=──┬─FULL────┬─┘
                                            └─PARTIAL─┘

   ┌─,IDTYPE=RESOURCE──────┐  ┌─,SAVE=YES──────┐
►──┤                       ├──┤                ├──────────────────────┤
   └─,IDTYPE=──┬─CP────────┬─┘  └─,SAVE=──┬─NO──┬─┘
               ├─SSCP──────┤              └─YES─┘
               └─RESOURCE──┘
```

IO Trace Operands:

```
├──,ID=node_name──┬──────────┬──┬─,IDTYPE=RESOURCE──────┬──┬─,SAVE=YES──────┬──────────┤
                  └─,EVERY───┘  │                       │  │                │
                                └─,IDTYPE=──┬─CP────────┬─┘  └─,SAVE=──┬─NO──┬─┘
                                            ├─SSCP──────┤              └─YES─┘
                                            └─RESOURCE──┘
```

LINE Trace Operands:

```
                    ┌─,COUNT=ALL (3)──────────────┐
├──,ID=line_name────┤                             ├──────────────────────────┤
                    └─,COUNT=──┬─ALL──────────────┘
                               └─number_of_bytes──┘
```

SIT Trace Operands:

```
                    ┌─,COUNT=ALL (3)──────────────┐
├──,ID=line_name────┤                             ├──┬──────────────────────┬──┤
                    └─,COUNT=──┬─ALL──────────────┘  └─,TRACEPT=trace_point_id─┘
                               └─number_of_bytes──┘
```

STATE Trace Operands:

```
                    ┌─,IDTYPE=RESOURCE──────┐   ┌─,SAVE=YES──────┐
├──,ID=node_name────┤                       ├───┤                ├──────────────┤
                    └─,IDTYPE=──┬─CP────────┘   └─,SAVE=──┬─NO───┘
                                ├─SSCP──────┤             └─YES──┘
                                └─RESOURCE──┘
```

Notes:

1. Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.
2. Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.
3. COUNT applies only to the IBM 3720 and 3745 Communication Controllers.

Specifies whether VTAM starts or cancels a specific type of trace for a node or line, or monitors the usage of all the VTAM buffer pools.

If you code the TRACE option, code the TYPE qualifier immediately following TRACE.

You can specify a network-qualified name on the ID=*node_name* operand of the TRACE start option for buffer and I/O traces.

You can change the value of TRACE with the MODIFY TRACE command while VTAM is running.

AMOUNT
Specifies whether VTAM records part or all of the data that is transmitted in inbound and outbound message buffers.

AMOUNT=PARTIAL
Specifies that VTAM records the data in trace records with a maximum size of 256 bytes. Each trace record contains a trace record header and data. Data that does not fit in the trace record is lost. This operand can be abbreviated as AMT=P.

AMOUNT=FULL
Specifies that VTAM records all of the data transmitted in message buffers.

Multiple trace records might be needed to record all of the data. This operand can be abbreviated as AMT=F.

COUNT=*number_of_bytes* **or ALL**
The number of bytes of data you want traced. The range is 0–254. If you do not code COUNT, or if you code COUNT=ALL, VTAM traces a maximum of 255 bytes of data in the PIU.

COUNT is valid only if you are running an SIT or LINE trace for a 3720, 3725, or 3745 communication controller.

EVERY
Specifies that VTAM starts or stops the requested trace for the resource named in the ID operand and all appropriate minor nodes of this resource.

You can abbreviate this operand as E.

This operand applies only to buffer and I/O traces. For example:

```
TRACE,TYPE=IO,ID=line name,E
```

This start option initiates an I/O trace for the line and each of its minor nodes.

This operand is not valid if ID=ISTPUS or ID=ISTIRN is also coded. If ID names a channel-attachment major node, code the EVERY option. You can name the link in a channel-attachment major node, but you cannot name the link station.

IDTYPE
Specifies the type of resource that the ID operand names. In cases where several types of resources share the same name, IDTYPE can be used to identify which resource the start option should act on.

If both a CP and an SSCP have the name specified on the ID operand, then both resources are traced.

IDTYPE is valid only for the buffer contents, I/O, and resource state traces.

IDTYPE=CP
Specifies that the name on the ID operand is a CP. Only the trace for a host CP or for the cross-domain resource that represents the adjacent CP is affected by this start option.

IDTYPE=SSCP
Specifies that the name on the ID operand is an SSCP. Only the trace for an SSCP is affected by this start option.

IDTYPE=RESOURCE
Specifies that the name on the ID operand is one of the following. VTAM looks for the resource in the order they are listed here:
1. SSCP
2. Host CP
3. Adjacent CP
4. Another resource

ID=line name
Names the specific line for which a VTAM trace is started or stopped. *line name* corresponds to the name on the LINE statement that represents the same line.

Each line traced must be explicitly coded in a TRACE start option.

ID=node name

Names the specific node for which a VTAM trace is started or stopped. *node name* corresponds to the name assigned to the node and filed in the VTAM definition library. *node name* can be a network-qualified name for the buffer and I/O traces. Refer to the description of the MODIFY TRACE command in *OS/390 SecureWay Communications Server: SNA Operation* for a list of the types of nodes you can choose.

For CMIP application programs, *node_name* is the name of the internal or external application program. For the VTAM topology agent, *node_name* is ISTTOPAG. For notification services, *node_name* is ISTNOTIF.

For external CMIP application programs, *node_name* is the application name defined as the ACB name of the application program major node.

For a CDRM, you can specify a network-qualified name, but this does not remove the restriction that the non-network-qualified CDRM name must be unique across networks.

You can use ID=VTAM to trace all SSCP activity and, in a multiple-domain network, all cross-domain resource manager activity. You can use ID=ISTPUS or the HOSTPU value to trace all SSCP-PU activity. Each terminal or logical unit traced must be explicitly coded in a TRACE start option.

NOTRACE

Specifies that VTAM cancels the trace requested by the TYPE operand. You should code this start option only when it is necessary to override a predefined TRACE start option. Code a separate NOTRACE start option to stop each trace.

SAVE

Applies to TYPE=BUF, TYPE=IO, and TYPE=STATE. It specifies whether the trace request is saved for the resource named on the ID operand.

SAVE=YES

VTAM saves the trace request for the resource named on the ID operand. If the resource exists when this request is issued, the trace starts immediately. If the resource does not exist when this request is issued, VTAM saves the trace command and starts the trace when the resource is defined.

For TYPE=BUF, TYPE=IO, and TYPE=STATE, if you start a trace with IDTYPE=RESOURCE and a CP or an SSCP (but not both) exists when the command is issued, VTAM starts the trace for the existing resource and saves the trace commands for both resources. If both a CP and an SSCP exist when the command is issued, VTAM starts tracing for them immediately.

Use the MODIFY NOTRACE command to delete a saved trace command. VTAM will not delete a saved trace command until you issue a MODIFY NOTRACE command for it, even though the resource might be created and freed or activated and deactivated several times. Saved trace commands are lost when VTAM is halted and restarted.

SAVE=NO

Does not save the trace request. If the resource does not exist when you start the trace, the request fails.

TRACE
Specifies that VTAM starts a specific type of trace for a node or line, or monitors the use of all the VTAM buffer pools. Once started, the trace remains in effect until it is stopped, either by stopping VTAM or by entering the MODIFY NOTRACE command. More than one trace can run concurrently, but code a separate TRACE start option to start each trace.

TRACEPT=trace_point_id
Specifies the point in the microcode at which tracing should be activated. TRACEPT applies to TYPE=SIT and is valid only if you are tracing CSS resources on an IBM 3745 Communication Controller. If you omit this operand, tracing is done for all valid trace points. The TRACEPT operand allows you to limit the tracing to a single trace point if too much output is being produced.

VTAM accepts any integer in the range 1–255; however, only a few values are defined by the NCP. For information on which values are defined and what they mean, refer to the *NCP, SSP, and EP Trace Analysis Handbook*.

TYPE=BUF
Specifies whether a VTAM buffer trace is started or stopped.

TYPE=IO
Specifies whether a VTAM I/O trace is started or stopped.

TYPE=LINE
Specifies that an NCP line trace is started or stopped for a communication controller. There can be up to 8 concurrent line traces and scanner interface traces (SIT) for each NCP, depending on how each NCP is generated.

TYPE=SIT
Specifies that a scanner interface trace is started or stopped for a 3720, 3725, or 3745 communication controller. There can be up to 8 concurrent line traces and SITs for each NCP, depending on how each NCP was generated.

TYPE=SMS,ID=VTAMBUF
Specifies that VTAM monitors the number of its requests to obtain buffers in the VTAM buffer pools. (After a specified number of requests occur, the trace creates a record to show how the pools are being used at that time.) If you code TYPE=SMS, also code ID=VTAMBUF.

TYPE=STATE
Specifies whether a resource state trace is started or stopped to monitor changes in the state of the resource named on the ID operand.

TRACE for MODULE, STATE (with OPTION), or VTAM Internal Trace

```
                        (1)
   ┌NOTRACE,TYPE=MODULE──────────────────┐
►►─┤                                     ├──────────────────────────────►
   │                                (2)  │
   └TRACE,TYPE=MODULE,┤ MODULE Operands ├┘

   ┌NOTRACE,TYPE=STATE (1)───────────────┐
►──┤                                     ├──────────────────────────────►
   └TRACE,TYPE=STATE┤ STATE Trace Operands (2) ├┘
```

```
►─┬─TRACE,TYPE=VTAM,─┬─ VIT Operands (2) ─┬─────────────────────────────────►◄
  │                  │                    │
  │                  │       (3)          │
  └─NOTRACE,TYPE=VTAM──────────────────────┘
```

MODULE Operands:

```
├──,OPTION=──┬─ALL──────────────────┬──────────────────────────────────────┤
             ├─option───────────────┤
             ├─(option)─────────────┤
             └─(──┬─COMMAND────────┬─)─┘
                  ├─CONNECTIONS────┤
                  ├─DEFINITION─────┤
                  ├─INTERFACES─────┤
                  ├─MANAGEMENT─────┤
                  ├─NOEXIT─────────┤
                  └─SESSION────────┘
```

STATE Trace Operands:

```
├──,OPTION=──┬─ALL──────────────┬──────────────────────────────────────┤
             ├─option───────────┤
             ├─(option)─────────┤
             └─(──┬─ADJCP─────┬─)─┘
                  ├─APPL──────┤
                  ├─CDRM──────┤
                  ├─CDRSC─────┤
                  ├─GROUP─────┤
                  ├─LINE──────┤
                  ├─LU────────┤
                  ├─NCP───────┤
                  └─PU────────┘
```

VIT Operands:

```
├─┬─,MODE=INT,SIZE=100,DSPSIZE=0──────────────────────────┬──────────────►
  │                                                        │
  └─,MODE=──┬─EXT──┬─,BFRNUM=2──────┬──────────────────────┤
            │      └─,BFRNUM=number─┘                      │
            └─INT──┬──────────────────────────────────────┤
                   └─,SIZE=size─┘   └─,DSPSIZE=dspsize─┘
```

Start Options

```
                                                  (4)  (5)
  ┌─,OPTION=(API,CIO,MSG,NRM,PIU,PSS,SMS,SSCP)──────────┐
►─┼──────────────────────────────────────────────────────┼──────────────────────┤
  ├─,SUBTRACE=ARBP,OPTION=─┬─HPR──────────────────────────┤
  │                        ├─ALL──────────────────────────┤
  │                        │                   (6)        │
  │                        └─(..,HPR,..)──────────────────┤
  ├─,SUBTRACE=TREE,OPTION=─┬─SSCP─────────────────────────┤
  │                        ├─ALL──────────────────────────┤
  │                        │                   (7)        │
  │                        └─(..,SSCP,..)─────────────────┤
  └─,OPTION=─┬─ALL──────────────────────────────────────┤
             ├─NONE─────────────────────────────────────┤
             ├─option───────────────────────────────────┤
             ├─(option)─────────────────────────────────┤
             │      ┌──,──────────┐
             │      ▼             │
             └─(────┬─API──────────┴──────────)──────┤
                    ├─APPC─┤
                    ├─CFS──┤
                    ├─CIA──┤
                    ├─CIO──┤
                    ├─CMIP─┤
                    ├─CSM──┤
                    ├─ESC──┤
                    ├─HPR──┤
                    ├─LCS──┤
                    ├─LOCK─┤
                    ├─MSG──┤
                    ├─NRM──┤
                    ├─PIU──┤
                    ├─PSS──┤
                    ├─SMS──┤
                    ├─SSCP─┤
                    ├─TCP──┤
                    ├─VCNS─┤
                    ├─XBUF─┤
                    └─XCF──┘
```

Notes:

1 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

2 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

3 NOTRACE,TYPE=VTAM is accepted but ignored. Tracing is started with the default trace table size and the default options.

4 The default options apply only to MODE=INT.

5 PSS and SMS can be turned off.

6 If multiple trace options are coded in parentheses, HPR must be one of the options coded inside the parentheses when SUBTRACE=ARBP is coded.

7 If multiple trace options are coded in parentheses, SSCP must be one of the options coded inside the parentheses when SUBTRACE=TREE is coded.

Specifies whether VTAM starts or cancels a specific type of trace.

If you code the TRACE option, code the TYPE qualifier immediately following TRACE.

You can change the value of TRACE with the MODIFY TRACE command while VTAM is running.

BFRNUM=*number*
Specifies the number of 8K external trace buffers the VTAM internal trace is to allocate and use for generalized trace facility (GTF) processing. Integers 0 or 2–50 can be specified. The default is 2. Two times the number of processors in the central processing unit (CPU) should ensure that enough buffers are available.

When an integer of 2–50 is specified, VTAM accumulates approximately 8K of external trace data prior to sending the data to GTF.

If zero is specified or there is no buffer available for the trace record, VTAM sends each trace record to GTF as it is recorded. This incurs a significant system overhead, and should be avoided.

DSPSIZE
Controls the use of the VIT data space. The VIT data space can store much more of the internal VIT than is possible using only the fixed CSA internal VIT. The default DSPSIZE is 0, meaning the data space will not be used. Valid values are 0 through 5, where a value of 1 is equal to 10 megabytes, a value of 2 is equal to 20 megabytes, and so on.

MODE
Specifies that the VTAM internal trace is to record its data on an internal, wraparound table (MODE=INT) and optionally, on an external trace file (MODE=EXT).

MODE applies only to TYPE=VTAM.

You can record trace data internally and externally simultaneously. If desired, you can have different sets of trace options active for internal and external recording. VTAM always runs with MODE=INT and the default trace options, whether you request tracing or not.

You must run specific operating system utilities to trap, format, and view external trace output. See *OS/390 SecureWay Communications Server: SNA Diagnosis* for more information about use of these operating system utilities.

Do not specify MODE=EXT and SIZE on the same command.

MODE=INT
Specifies that the VTAM internal trace is to record its data on an internal, wraparound table.

MODE=EXT
Specifies that the VTAM internal trace is to record its data on an external trace file as well as on an internal, wraparound table. You can record external trace data using the generalized trace facility (GTF). GTF must be active when you initiate traces. You can format output using IPCS.

Refer to *OS/390 SecureWay Communications Server: SNA Diagnosis* for a more detailed description of formatting and printing trace output.

NOTRACE
Specifies that VTAM cancels the trace requested by the TYPE operand. Code this start option only when it is necessary to override a predefined TRACE start option. Code a separate NOTRACE start option to stop each trace.

NOTRACE,TYPE=VTAM
Accepted but ignored. Tracing is started with the default trace table size and the default options.

OPTION=(option,option,...option)
Applies to TYPE=MODULE, TYPE=STATE and TYPE=VTAM.

You can abbreviate this operand as OPT.

You can code one or more of the options listed for each TYPE operand.

Note: Each trace option can affect the performance of your host and the VTAM network. Be sure to understand the effect this function can have on performance in your environment before specifying OPTION=ALL.

For **TYPE=MODULE**, OPTION specifies the modules for which tracing is to be started.

For **TYPE=STATE**, OPTION specifies the types of resources for which resource states are to be recorded. The data is recorded using the mode (internal or external) specified for the SSCP VIT option.

For **TYPE=VTAM**, OPTION is a listing of the VTAM functions that you want to trace.

When VTAM encounters an exception condition involving an API, NRM, PIU, SSCP, or SMS function, it generates a trace record whether or not you have activated that option.

Notes:

1. Although the default options are always active for internal recording, these options do not appear in DISPLAY TRACES output unless you have specified them on the MODIFY TRACE command or the TRACE,TYPE=VTAM start option.
2. If you start VTAM with INOPDUMP=ON, the CIO option is traced. Unlike the five default options (API, MSG, NRM, PIU, SSCP), the user can control the use of the CIO option.

ALL
Applies to TYPE=MODULE, TYPE=STATE and TYPE=VTAM.

For **TYPE=MODULE**, OPTION=ALL starts the tracing of all the modules shown on the OPTION operand for which TYPE=MODULE apply.

For **TYPE=STATE**, OPTION=ALL starts the tracing of resource states for all of the resource types shown on the OPTION operand for which TYPE=STATE apply.

For **TYPE=VTAM**, OPTION=ALL specifies that the VTAM internal trace (VIT) is to be started for all of the VTAM internal functions for which the VIT is available, including VCNS trace functions. It is equivalent to specifying all of the internal trace types. Exception trace entries continue to be recorded.

For **TYPE=MODULE**, you can also specify the following options:

OPTION=COMMAND
Starts tracing modules involved in command processing.

OPTION=CONNECTIONS
Starts tracing modules involved in setting up connections between nodes.

OPTION=DEFINITION
Starts tracing modules involved in resource definition processing.

OPTION=INTERFACES
Starts tracing modules involved in the interface with the host SSCP or the host CP.

OPTION=MANAGEMENT
Starts tracing modules involved in network management.

OPTION=NOEXIT
Specifies that module exits are not traced for modules associated with other OPTION values for TYPE=MODULE. Module exits are not traced for any modules until a subsequent MODIFY NOTRACE,TYPE=MODULE,OPTION=NOEXIT command is issued.

Starting a module trace for any OPTION with TYPE=MODULE starts the tracing of the module exits also, unless you specify OPTION=NOEXIT.

OPTION=SESSION
Starts tracing modules involved in session establishment.

For **TYPE=STATE**, you can also specify the following options:

OPTION=ADJCP
Starts tracing the states of all adjacent control points.

OPTION=APPL
Starts tracing the states of all application programs.

OPTION=CDRM
Starts tracing the states of all CDRMs.

OPTION=CDRSC
Starts tracing the states of all CDRSCs.

OPTION=GROUP
Starts tracing the states of all line groups.

OPTION=LINE
Starts tracing the states of all lines.

OPTION=LU
Starts tracing the states of all logical units.

OPTION=NCP
Starts tracing the states of all NCPs.

OPTION=PU
Starts tracing the states of all physical units.

For **TYPE=VTAM**, you can also specify the following options:

API
Helps you determine whether an application program is causing a problem. API entries are written for RPL macros, RPL exit routines, user exit routines, and user posts.

APPC
Helps you determine whether an LU 6.2 application is causing a problem. LU 6.2 entries are written for the following:
APPCCMD macro invocations
User posts and exit scheduling by LU 6.2 code
Calls to a security manager for security processing
Message unit transmissions between LU 6.2 components

CFS
Helps you assess and manage your coupling facility structure.

CIA
Helps isolate problems indirectly related to channel IO by tracing additional data near the time of channel IO activity.

CIO
Helps isolate problems related directly to channel I/O (CIO). CIO entries are written for attentions, error recovery, interruptions, HALT I/O SVC, and START I/O SVC.

CMIP
Indicates whether a CMIP problem is in VTAM or in a CMIP application program.

This option traces:
- Calls from CMIP application programs to the management information base (MIB) application programming interface
- Calls to the read-queue exit of the CMIP application program
- Topology updates from VTAM resources

CSM
Traces the parameter list information that flows across the communications storage manager (CSM) interface for the following events:
- Requests to obtain a buffer
- Requests to free a buffer
- Requests to assign additional tokens to the buffer
- Requests to change the ownership of a buffer
- Requests to change the pageable state of a buffer
- Requests to copy data to or from a CSM buffer or a user data area
- Buffer pool expansion
- Buffer pool contraction

Note: For installations using high performance data transfer (HPDT), the use of the CSM trace should be restricted to times when diagnostic information is needed. Using this option on a regular basis minimizes the performance benefits provided by HPDT services.

ESC
Helps you track, in detail, the flow of requests for a given process.

HPR
Helps you isolate problems related to high performance routing (HPR). Events include invocation of the HPRCTL macroinstruction, invocation of the HPR timer control macroinstruction, any LU-LU session data sent

across a rapid transport protocol (RTP) connection, a network layer packet (NLP) sent or received, a dispatch of the RTP context manager (RCM) PAB, a dispatch of a RTP PAB, an error detected by RTP, and a route setup sent or received.

LCS
Helps you isolate problems occurring during data transfers from an IBM 3172 Interconnect Controller to VTAM. The LCS option enables tracing of data that VTAM receives from an IBM 3172 Interconnect Controller at four levels: LCSX (cross-channel), LCSP (port or adapter), LCSS (service access point of the adapter) and LCSL (line).

LOCK
Helps you determine when VTAM modules get and release locks.

MSG
Helps you correlate other VIT internal entries with the console messages even if the console sheet is lost. MSG entries are written for all messages to the VTAM operator.

NONE
Specifies that no traces are active.

NRM
Helps you follow the services of the network resource management component. These include the assignment of, references to, and the deletion of certain VTAM resources such as node names, network addresses, and control blocks. NRM entries are written for all I/O and buffer contents traces.

PIU
Helps you isolate problems to hardware, to the NCP, or to VTAM. PIU entries are written for all I/O to and from VTAM, which can be an advantage over I/O and buffer content traces.

PSS
Helps you track the flow of requests through VTAM. PSS entries are written for the VTAM macros that invoke and control PSS, and for scheduling and dispatching VTAM routines.

SMS
Helps you isolate problems caused by storage shortages. When used with the SSCP or PSS trace options, it can also help you isolate internal VTAM problems. SMS entries are written when SMS macros are used to request or release fixed- or variable-length buffers. SMS entries are also written when VTAM expands or attempts to expand a buffer pool.

SSCP
Helps you isolate problems to a specific VTAM component or module. SSCP entries are written for the request or response units (RUs) sent between VTAM components.

TCP
Helps you trace the communication between VTAM and TCP/IP. Events include socket API calls and completions, IUCV calls and completions, and invocations of the SOCTREE macro which controls updating a binary tree of socket control blocks. This option is only used with the VTAM AnyNet Feature.

VCNS
Helps you determine whether a VCNS application is causing a problem.

VCNS entries are written for VCNSCMD macro invocations, user posts, and exit scheduling by VCNS code, and work element transmissions between VCNS components.

XBUF
Records information contained within the extended buffer list, particularly in regard to information on CSM usage by VTAM. This option traces the contents from the application supplied extended buffer list as well as the internal buffer list that VTAM uses to carry the extended buffer list information.

Note: For installations using high performance data transfer (HPDT), the use of the XBUF trace should be restricted to times when diagnostic information is needed. Using this option on a regular basis minimizes the performance benefits provided by HPDT services.

XCF
Helps you trace XCF communication.

SIZE=*size*
The number of pages in the internal trace table. SIZE is valid only when MODE=INT is specified. *size* is a decimal integer in the range 100–999. The default is 100, and if you specify a value less than 100, 100 is used.

If you do not choose a large enough value for the SIZE option, valuable information can be destroyed because of wraparound in the trace table. If you use this operand to select a different size on a subsequent MODIFY TRACE command, information is lost because VTAM frees the trace table so that it can obtain another table with a new size.

SUBTRACE
Specifies that SUBTRACE can be used to turn on a subset of trace entries under a trace option. Currently, two SUBTRACE types are defined; subtrace TREE is defined under the SSCP trace option and subtrace ARBP is defined under the HPR trace option.

Note: Both of the SUBTRACE options are default to off and they could be used for diagnostic purposes. It is not recommended to activate them at VTAM start time.

SUBTRACE=ARBP
Specifies the ARB (Adaptive Rate Based) algorithm performance trace records that are to be generated. The OPTION operand is required when SUBTRACE is specified and HPR must be one of the trace options specified when SUBTRACE=ARBP is coded. After subtrace ARBP is activated, the following trace records will be generated for the ARB algorithm processing: ARBR (Generated when ARB Responsive Mode algorithm is utilized) and ARBB (Generated when ARB Base Mode algorithm is utilized).

SUBTRACE=TREE
Specifies the APPN routing tree trace records that are to be generated. The OPTION operand is required when SUBTRACE is specified and SSCP must be one of the trace options specified when SUBTRACE=TREE is coded. After subtrace TREE is activated, the following trace records will be generated for routing trees used by APPN route computation: TRRT, TRR2, TRR3, TRR4, TRR5, HLST, and HLS2.

TYPE=MODULE
Specifies whether a module trace is started or stopped for the types of modules listed on the OPTION operand.

TYPE=STATE
Specifies whether a resource state trace is started or stopped to monitor changes in the state of the type of resources listed on the OPTION operand.

TYPE=VTAM
Specifies the VTAM internal trace (VIT).

TRANSLAT

```
   ┌─TRANSLAT=(0,1,2,3,4,5,6,7)──────────┐
►►─┼─────────────────────────────────────┼──────────────────────────────────────►◄
   └─TRANSLAT=──┬─value──────────────────┬┘
                ├─()─────────────────────┤
                └─(──┬─────────────────┬──)─┘
                     ├─┬─OLUALIAS─┬─┤
                     │ └─0────────┘ │
                     ├─┬─DLUREAL──┬─┤
                     │ └─1────────┘ │
                     ├─┬─DLUSSCP──┬─┤
                     │ └─2────────┘ │
                     ├─┬─COSNAME──┬─┤
                     │ └─3────────┘ │
                     ├─┬─LOGMODES─┬─┤
                     │ └─4────────┘ │
                     ├─┬─LUANAME──┬─┤
                     │ └─5────────┘ │
                     ├─┬─USERVAR──┬─┤
                     │ └─6────────┘ │
                     └─┬─DLUALIAS─┬─┘
                       └─7────────┘
```

Controls alias name translation. It specifies which names will be translated, but not how the names will be translated. If TRANSLAT is specified, only the bits corresponding to the values specified will be turned on. All other bits will be turned off. For example, if you specify

```
TRANSLAT=(OLUALIAS,DLUSSCP,LOGMODES)
```

the hexadecimal value will be X'A8' (equivalent to B'10101000'). If you specify TRANSLAT=(), all bits will be turned off.

TRANSLAT=OLUALIAS or 0 (bit 0)
Specifies that VTAM is to determine the originating logical unit (OLU) alias names.

TRANSLAT=DLUREAL or 1 (bit 1)
Specifies that VTAM is to determine destination logical unit (DLU) real names.

TRANSLAT=DLUSSCP or 2 (bit 2)
Specifies that VTAM is to determine the DLU's owning SSCP name.

TRANSLAT=COSNAME or 3 (bit 3)
Specifies that VTAM is to determine class-of-service names.

TRANSLAT=LOGMODES or 4 (bit 4)
Specifies that VTAM is to determine logon mode names.

TRANSLAT=LUANAME or 5 (bit 5)
Specifies that VTAM is to determine associated LU alias names.

TRANSLAT=USERVAR or 6 (bit 6)
Specifies that VTAM is to enable the USERVAR function.

TRANSLAT=DLUALIAS or 7 (bit 7)
Specifies that VTAM is to determine DLU alias names only when needed.

UPDDELAY

```
                (1)
  ┌─UPDDELAY=60──────┐
►►┼                  ├──────────────────────────────────────────►◄
  └─UPDDELAY=max_time─┘
```

Notes:

1 UPDDELAY is meaningful only if the OSIMGMT=YES start option is also used.

range: 5–65535 seconds (or equivalent value in minutes or hours)

Specifies the maximum time the VTAM topology agent waits between checking the list of resources in transient states (such as PENDING ACTIVE and PENDING INACTIVE) to determine whether the resources have been in transient states long enough to be sent to the topology manager. If a resource reaches resting state before the interval set by UPDDELAY has expired, the update for that resource is still sent.

By setting a longer interval on the UPDDELAY start option, you can reduce the CPU cycles required to send updates to the topology manager. By setting a shorter interval, you can send updates to the topology manager more frequently. Frequent sending of updates increases the opportunity to discover resources that are hung in transient states.

You can change the value of UPDDELAY with the MODIFY VTAMOPTS command while VTAM is running. If you change the value, the new value takes effect after the time specified on the original value has expired. For example, if the original value for UPDDELAY is 1 hour and you specify 5 seconds for the UPDDELAY start option on the MODIFY VTAMOPTS command, the new value of 5 seconds takes effect after the hour has expired.

The value for UPDDELAY can be expressed as:

n or nS
Specifies the maximum number of seconds between the range 5–65535.

nM
Specifies the maximum number of minutes between the range 1–1092.

nH
Specifies the maximum number of hours between the range is 1–18.

USSTAB

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─USSTAB=table_name─┘
```

Specifies the name of a USS table to be used for VTAM operator messages and for VTAM operator commands that are processed through USS. *table_name* is the 8-character name of a USS table. Choose one of the following USS tables:

- ISTINCNO, which contains the IBM-supplied VTAM messages and commands. ISTINCNO is the default for IBM-supplied VTAM messages. This table can be supplemented with a user-defined message table. When a user-defined USS command table is specified on the USSTAB start option, the user-defined table is the only table searched to process a VTAM command.
- ISTCFCMM contains a master copy of the original messages in ISTINCNO. Choose this table name if you want to continue using these original messages in program operator applications after ISTINCNO is modified. Commands processed through USS continue to use ISTINCNO.
- A user-defined table that contains user-modified messages and commands. Messages not defined in the user-defined table continue to use ISTINCNO; commands not defined in the user-defined table will not be found. The user-defined table should not be named either ISTINCNO or ISTINCDT; otherwise, you might not be able to use all of the defined USS commands and messages.

If the attempt to load the specified table fails, VTAM uses ISTINCNO by default. Message IST116I is issued if the specified table is not found; message IST448I is issued if storage is insufficient for loading the table.

VERIFYCP

```
                    (1)
    ┌─VERIFYCP=NONE──────────┐
►►──┤                        ├──────────────────────────────────────►◄
    └─VERIFYCP=──┬─NONE─────┬┘
                 ├─OPTIONAL─┤
                 └─REQUIRED─┘
```

Notes:

1 The VERIFYCP start option is meaningful only if the NODETYPE start option is also used.

Specifies whether VTAM performs session-level verification during activation of LU 6.2 sessions involving control points. The related SECLVLCP start option values specify the type of verification.

If you do not code SECLVLCP, VTAM attempts to use the enhanced protocol if VERIFYCP=OPTIONAL or VERIFYCP=REQUIRED is specified, but will accept basic protocol.

If you code VERIFYCP=OPTIONAL or VERIFYCP=REQUIRED, a security management product, such as RACF 1.9 or later, must be installed and active.

VERIFYCP=NONE
Specifies that there is no verification of the partner CP's identity during session activation.

VERIFYCP=OPTIONAL
Specifies that identity verification is done for certain partner CPs during session activation. Identity verification is done when there is a password defined for the CP partners in the installed security manager product. Refer to *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about how VTAM determines when to perform identity verification.

VERIFYCP=REQUIRED
Specifies that VTAM verifies the identity of all partner CPs during activation of sessions between LU 6.2 applications. Every partner CP must have a password defined. Any partner CPs that do not have a password defined cannot establish LU 6.2 sessions with this application program.

VFYRED

```
  ┌─VFYRED=YES──────────┐
►►┼─────────────────────┼──────────────────────────────────────────►◄
  └─VFYRED=──┬─YES─┬────┘
             └─NO──┘
```

dependencies: LU 6.2 application programs using APPCCMD API

Specifies whether resource verification reduction is allowed for all LU 6.2 application programs using the APPCCMD application program interface for this VTAM.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information about resource verification reduction support.

You can change the value of VFYRED with the MODIFY VTAMOPTS command while VTAM is running.

VFYRED=YES
Indicates that LU 6.2 session initiation requests do not need to be delivered to the target LU.

Note: If the BIND fails, VTAM reissues the session initiation request indicating that the resource must be verified.

VFYRED=NO
Indicates that the session initiation request must be delivered to the target LU. Normal session setup logic will be performed.

VFYREDTI

```
                 (1)
  ┌─VFYREDTI=OFF──────────────────┐
►►┼───────────────────────────────┼──────────────────────────────────►◄
  └─VFYREDTI=──┬─OFF──────────────┬┘
               ├─0────────────────┤
               └─reduction_timer──┘
```

Notes:

1 VFYREDTI is meaningful only if the NODETYPE=NN start option is also used.

dependencies:	LU 6.2 application programs using the APPCCMD API
range:	1–2147483647 seconds (or equivalent minutes or hours)

Specifies the maximum amount of time a resource's location is not verified during session setup.

You can change the value of VFYREDTI with the MODIFY VTAMOPTS command while VTAM is running.

VFYREDTI=OFF
Indicates that there is no node-wide verification reduction timer value. Verification reduction will be attempted for the time indicated by the target resource's cached information. If the target resource has no cache time value VTAM will assume a time value of 1 day.

VFYREDTI=0
Indicates that all requests must be verified.

VFYREDTI=*reduction_timer*
Specifies the amount of time resource verification reduction is to be performed before the resource is verified. The lesser of this value and the target resource's cache time value (or the 1 day default) is used.

reduction_timer can be specified in seconds (S), minutes (M), or hours (H), where n is an integer:

n or nS
specifies the number of seconds resource verification reduction is to be performed before the resource is verified. Seconds is the default interval.

nM Specifies the number of minutes resource verification reduction is to be performed before the resource is verified.

nH Specifies the number of hours resource verification reduction is to be performed before the resource is verified.

VRTG

```
                (1)
   ┌─VRTG=NO──────────┐
►►─┤                  ├──────────────────────────────────────────────────►◄
   └─VRTG=──┬─YES─┬──┘
            └─NO──┘
```

Notes:

1 VRTG is meaningful only if the NODETYPE and HOSTSA start options are also used.

Indicates whether VR-based transmission group connections are to be requested when the SSCP-SSCP session is established for this node.

This option is valid only for interchange nodes and migration data hosts when start options NODETYPE and HOSTSA are specified.

Note: This value can be overridden by coding the VRTG operand in the CDRM definition statement.

You can change the value of VRTG with the MODIFY VTAMOPTS command while VTAM is running.

VRTG=YES
Indicates that VR-based transmission group connection is requested when SSCP-SSCP sessions are established.

VRTG=NO
Indicates that virtual-route-based transmission group connection is not requested when SSCP-SSCP sessions are established.

VRTGCPCP

```
                  (1)
   ┌─VRTGCPCP=YES──────┐
►►─┤                   ├──────────────────────────────────────────────────►◄
   └─VRTGCPCP=─┬─YES─┬─┘
               └─NO──┘
```

Notes:

1 VRTGCPCP is meaningful only if the NODETYPE and HOSTSA start options are also used.

dependencies: VRTG=YES coded

Indicates whether CP-CP sessions are supported over the virtual-route-based transmission group.

This option is valid only for interchange nodes and migration data hosts when start options NODETYPE and HOSTSA are specified.

Note: This value can be overridden by coding the VRTGCPCP operand in the CDRM definition statement.

You can change the value of VRTGCPCP with the MODIFY VTAMOPTS command while VTAM is running.

VRTGCPCP=YES
Indicates that CP-CP sessions are supported over the VR-based transmission group.

VRTGCPCP=NO
Indicates that CP-CP sessions are not supported over the VR-based transmission group.

VTAMEAS

```
   ┌─VTAMEAS=32001──────────────────────────┐
►►─┤                                        ├──────────────────────────────►◄
   └─VTAMEAS=number_of_concurrent_sessions─┘
```

range: 0–2147483647

Specifies the number of concurrent sessions VTAM can have with other LUs.

XCFINIT

```
►►─┬─XCFINIT=YES──────────┬──────────────────────────────────────────►◄
   │                      │
   └─XCFINIT=──┬─NO──┬────┘
               └─YES─┘
```

dependencies: OS/390; applies only to VTAMs defined in a sysplex environment; HPR=RTP ; NODETYPE=EN and NODETYPE=NN

Specifies whether XCF connections to other VTAMs in a sysplex are to be established at VTAM initialization.

XCFINIT=NO
Specifies that XCF connections to other VTAMs are not to be established at VTAM initialization.

XCFINIT=YES
Specifies that XCF connections to other VTAMs are to be established at VTAM initialization.

XNETALS

```
►►─┬─XNETALS=NO───────────┬──────────────────────────────────────────►◄
   │                      │
   └─XNETALS=──┬─NO──┬────┘
               └─YES─┘
```

dependencies: NETID on PU statement

Determines whether the network ID of an attaching node or VTAM's network ID is used for an adjacent nonnative node. Used to control how a nonnative node can attach to VTAM.

Note: The XNETALS option has changed in the way VTAM treats PUs that send in non-native NETIDs in their XID. VTAM used to let the PU connect and treated it as if it sent in the same NETID that VTAM started with but that is no longer true.

Note: Coding XNETALS=NO and GWCAP=NO will prevent subarea searches of the node in which the options are coded when the DLU NETID is not the NETID of the host. If YES is coded for either option, subarea searches will be performed for the host in which the options were coded.

The following matrix shows the resulting XNETALS values when XNETALS and CONNTYPE are coded for various levels:

Table 66. XNETALS Matrix

Start List Values	Specified on GROUP, LINE, or PU (Lowest level wins)	CONNTYPE	XNETALS and Result for the PU
YES	YES	APPN	YES

Table 66. XNETALS Matrix (continued)

Start List Values	Specified on GROUP, LINE, or PU (Lowest level wins)	CONNTYPE	XNETALS and Result for the PU
YES	YES	LEN	YES
YES	YES		YES
YES	NO	APPN	NO
YES	NO	LEN	NO
YES	NO		NO
YES		APPN	YES
YES		LEN	YES
YES			YES
NO	YES	APPN	YES
NO	YES	LEN	YES
NO	YES		YES
NO	NO	APPN	NO
NO	NO	LEN	NO
NO	NO		NO
NO		APPN	NO
NO		LEN	NO
NO			NO
not specified	YES	APPN	YES
not specified	YES	LEN	YES
not specified	YES		YES
not specified	NO	APPN	NO
not specified	NO	LEN	NO
not specified	NO		NO
not specified		APPN	YES
not specified		LEN	NO
not specified			NO (Default)

XNETALS=NO

Indicates that VTAM's network ID will be used to represent the connecting node's network ID.

During connection:

- If the NETID operand is specified on the PU statement then, it must specify VTAM's network ID.
- If the NETID operand is not coded on the PU statement then, VTAM's network ID is assumed and the adjacent node's network ID is ignored.

During LU-LU session establishment:

- If the network ID for the originating LU is specified (during connection time) and is not identical to VTAM's network ID, the session will not be established.
- If the originating LU is not using a network qualified name, VTAM's network ID is assumed and the session will be established.

Note: If the PU is a dependent LU server or requester, nonnative network connection is not allowed if XNETALS=NO is specified and the NETID operand is not coded.

XNETALS=YES

Indicates that the network ID of the adjacent PU will be used for connection and session establishment.

During connection:

- If the NETID operand on the PU statement is coded, it will be used as the PU's network ID. In this case, the network ID specified on the PU statement must match the network ID specified by the attaching node during connection processing.
- If the NETID operand is not specified on the PU statement, then the network ID specified by the attaching node during connection processing will be assumed.

During LU-LU session establishment:

- If the network ID for the originating LU is specified and does not match the PU's network ID, the session will not be established.
- If the originating LU name is not a network qualified name, the session will be established. VTAM will qualify the LU's name with the attaching node's network ID.

Chapter 5. User-Defined Tables and Data Filter

User-Defined Tables and Data Filter

About This Chapter

This chapter describes how to create or modify the following tables and filter:
- Associated LU
- Subarea and APPN class-of-service (COS)
- CMIP services directory definition file
- Interpret
- Logon mode
- Message-flooding prevention
- Model name
- Subarea mapping (SAMAP)
- Session awareness (SAW) data filter
- USS

Follow the assembler language conventions described in “Format of Macroinstructions” on page 10 when coding the macroinstructions discussed in this chapter.

The MODIFY TABLE operator command lets you dynamically replace the associated LU, COS, interpret, logon mode, model name, and USS tables, the CMIP services directory definition file, and the session awareness (SAW) data filter. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on the MODIFY TABLE command.

When link editing a table for which IBM has shipped a default, it is recommended that you use the linkage editor default parameters.

To assemble the tables, your JCL will need to point to the SYS1.SISTMAC1 macroinstruction library.

Associated LU Table

An associated LU table contains associated LU names that can be passed to VTAM application programs in their LOGON exits. VTAM application programs use the associated LU names to create dynamic definitions for their session-partner resources. These names specify primary and alternate printers that are logically related to the SLU. This information is not available in model definitions. For information on model definitions, see “Model Name Table” on page 720.

Operands on an SLU’s resource definition can associate that SLU with the proper associated LU data. The ASLTAB operand specifies the associated LU table to be used, and the ASLENT operand specifies the proper entry within the table. These operands are described for each LU-type resource in Major Nodes.

© Copyright IBM Corp. 1984, 1999

Associated LU Table

Table 67. Associated LU Table Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	ASLTAB					
name	ASLENT	PRINTER1	663			
		PRINTER2	664			
name	ASLPLU	PLU	663	R		
		PRINTER1	663			
		PRINTER2	663			

Legend:

R Required operand.

Coding Notes

ASLTAB Macroinstruction:

The ASLTAB macroinstruction indicates the beginning of the associated LU table.

ASLENT Macroinstruction:

The ASLENT macroinstruction indicates the start of an associated LU table entry and optionally builds a default set of associated LU data. The end of the entry is indicated either by another ASLENT macroinstruction or by an end of file.

ASLPLU Macroinstruction:

The ASLPLU macroinstruction sets up associated LU data for a specific PLU within the table entry started by the previous ASLENT macroinstruction.

Installing an Associated LU Table

Store the set of macroinstructions as a member of SYS1.VTAMLST and use the member name as the name of the table.

No assembly or link-edit is required to install this table. The table will be dynamically built upon the first activation of any resource that has a defined association to the table. When the last resource using it is deactivated, the table is deleted and its storage is freed.

IBM does not supply a default associated LU table.

Full Syntax

The full syntax for the associated LU table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

ASLTAB:

```
►►──┬──────┬──ASLTAB──────────────────────────────────────────────────►◄
    └name──┘
```

ASLENT:

```
►►──name──ASLENT──┬──────────────────────────┬──┬──────────────────────────┬──────────────────────►◄
                  └PRINTER1=lu_name──┘       └PRINTER2=lu_name──┘
```

ASLPLU:

```
►►──┬──────┬──ASLPLU──PLU=plu_name──┬──────────────────────────┬──┬──────────────────────────┬──►◄
    └name──┘                        └PRINTER1=lu_name──┘       └PRINTER2=lu_name──┘
```

Operand Descriptions

The following section describes the operands you can code for the associated LU table.

name:

```
►►──┬──────┬──────────────────────►◄   ►►──name──────────────────────►◄
    └name──┘
```

macroinstructions: ASLTAB, ASLPLU *macroinstructions:* ASLENT

On ASLTAB and ASLPLU: Specifies optional names for the ASLTAB and ASLPLU macroinstructions.

On ASLENT: Specifies the required 1–8 character name for the ASLENT macroinstruction.

PLU:

```
►►──PLU=plu_name──────────────────────────────────────────────────────►◄
```

macroinstructions: *ASLPLU*
dependencies: *none*

Specifies the 1–8 character name of the PLU as it is known in the SLU's network.

If you omit PRINTER1 and PRINTER2, no corresponding associated LU name is sent to the PLU or the subsystem during session initiation. Omit the PRINTER1 and PRINTER2 names if you have specified default data by using the ASLENT macroinstruction, but make sure that you prevent any associated LU data from being sent to a specific PLU.

PRINTER1:

Associated LU Table

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └PRINTER1=lu_name──┘
```

macroinstructions: *ASLENT, ASLPLU*
dependencies: *none*

Specifies the 1–8 character name of the primary printer associated with the terminal as identified in the SLU's network.

PRINTER2:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └PRINTER2=lu_name──┘
```

macroinstructions: *ASLENT, ASLPLU*
dependencies: *none*

Specifies the 1–8 character name of the alternate printer associated with the terminal as identified in the SLU's network.

If you code PRINTER1 or PRINTER2, the printer must be in the same network as the terminal or must have a unique name that requires no translation across network boundaries. Violation of this restriction might cause failure of print-screen operations during cross-network sessions.

If you code PRINTER1 or PRINTER2 on the ASLENT macroinstruction, VTAM uses this value for any PLUs that are not specified in a following ASLPLU macroinstruction.

If both PRINTER1 and PRINTER2 are omitted on the ASLENT macroinstruction, no default data is created. Consequently, any PLU not specified in one of the ASLPLU macroinstructions within this entry will not normally receive any associated LU data.

Class of Service (COS)

You can define classes of service (COS) for subarea and APPN, though you do not need to create APPN classes of service unless your network has special requirements.

You define subarea COS tables with macroinstructions, and APPN COS with definition statements and operands. Details for defining classes of service follow in the next two sections.

Subarea Class-of-Service Table

To define subarea classes of service, create a class-of-service (COS) table with entries containing lists of routes grouped together on the basis of characteristics such as security, transmission priority, and bandwidth. The characteristics of a particular list determine a class of service. For example, you could place the fastest routes in one list for use in interactive sessions and slower routes in another list for use by batch jobs. If a session involves the transmission of sensitive data, you would place only secure routes in the list used for that session. By specifying the

name of an entry from the COS table in the logon mode table associated with a session, you can select the list of routes you want to be used for the session.

VTAM does not provide a default subarea COS table. However, you can define a default COS in the logon mode table by specifying ISTCOSDF on the LOGMODE operand of the MODEENT macroinstruction. You can modify the session characteristics in ISTCOSDF, though if you do, it is recommended that you put ISTCOSDF in a separate table rather than modifying it in the IBM-supplied default logon mode table, ISTINCLM.

VTAM also offers a default list of virtual routes that is used if you do not create a COS table. This default list is also used if you do not name a class of service in the logon mode table for a session. You can replace this default list by creating a COS table with an unnamed (blank) COS entry containing the new list. This new default list is then used if no COS entry is named in the logon mode table for a session.

In addition to coding an unnamed COS entry in your COS table, you might also want to code an entry for use in SSCP sessions (SSCP-SSCP, SSCP-PU, and SSCP-LU). You can specify the routes used for SSCP sessions by including an entry named ISTVTCOS in the COS table.

Table 68. Subarea COS Macroinstructions and Operands

Name	Macro-instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	COSTAB					
name	COS	VR	667	R		see description
		SUBSTUT	667		NO	
	COSEND					

Legend:

R Required operand.

Coding Notes

COSTAB Macroinstruction: The COSTAB macroinstruction begins the COS table and must precede all COS definition entries.

COS Macroinstruction: The COS macroinstruction defines a class-of-service entry.

COSEND Macroinstruction: The COSEND macroinstruction marks the end of the class-of-service table.

Note: You can also code an assembler end statement. However, if you code an assembler end statement, it must follow the COSEND macroinstruction.

Attention: Using assembler language statements in generating a COS table may cause an invalid COS table to be built.

Installing a Subarea COS Table

In a single-network installation, the assembled output of this process must be named ISTSDCOS. A VTAM acting as a gateway SSCP, designated to resolve COS

names for a gateway NCP, uses the COS table named on the NETWORK or BUILD definition statement in the gateway NCP's generation deck.

This table should be link-edited as a nonexecutable module into a library associated with the SYS1.VTAMLIB DD statement.

VTAM loads the resulting load module during initialization and uses it later to establish sessions.

Full Syntax

The full syntax for the subarea class of service table follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

COSTAB:

```
►►──name──COSTAB──────────────────────────────────────────────────────►◄
```

COS:

```
►►──┬──────┬──COS──VR=──┬─(virtual_route,trans_priority)──────────────┬──────►
    └─name─┘            └─(─(──virtual_route,trans_priority──)──────)─┘

    ┌─SUBSTUT=NO──────────┐
►───┤                     ├──────────────────────────────────────────────►◄
    │                 ┌─NO─┐
    └─SUBSTUT=────────┤    ├──┘
                      └─YES─┘
```

COSEND:

```
►►──COSEND────────────────────────────────────────────────────────────►◄
```

Operand Descriptions

The following section describes the operands you can code for the subarea class of service table.

name:

```
►►──name──────────────────────────►◄

►►──┬──────┬──────────────────────►◄
    └─name─┘
```

macroinstructions: COSTAB *macroinstructions:* COS

On COSTAB: Specifies a required 1–8 character name for the COSTAB macroinstruction. In a single-network installation, specify ISTSDCOS as the COSTAB *name*.

On COS: Specifies an optional 1–8 character name for the COS macroinstruction. If *name* is not specified, the unnamed default class of service is assumed. For SSCP sessions, specify ISTVTCOS as the COS *name*.

SUBSTUT:

```
►►─┬─SUBSTUT=NO──────┬──────────────────────────────────────────►◄
   │                 │
   └─SUBSTUT=─┬─NO──┬─┘
              └─YES─┘
```

macroinstructions: *COS*
dependencies: *none*

Indicates whether this entry will be substituted when VTAM does not recognize the COS name that is specified. SUBSTUT can be used for only one entry in the COS table. If more than one entry is coded, only the first one is recognized.

Note: You can monitor substitutions and reject establishment of sessions which you do not want to use the substitute COS values by coding a session management exit routine.

Refer to *OS/390 SecureWay Communications Server: SNA Customization* for more information on the session management exit routine, and to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more information on substituting class-of-service values.

VR:

```
►►─VR=─┬─(virtual_route,trans_priority)──────────┬──────────────────►◄
       └─(─(─virtual_route,trans_priority─)──)──┘
```

macroinstructions: *COS*
dependencies: *see rules following*

Specifies one or more ordered pairs of numbers that must meet the following rules:

- Each pair must be enclosed within parentheses and must consist of a virtual route number (*virtual_route*) and a transmission priority indicator number (*trans_priority*).
- If more than one pair is specified, the entire list must be enclosed within parentheses and each pair must be separated by a comma.
- A virtual route number must be a decimal integer in the range 0–7.
- A transmission priority indicator number must be a decimal integer in the range 0–2, where:

0 Specifies low-priority session traffic
1 Specifies medium-priority session traffic
2 Specifies high-priority session traffic

- You can specify up to 24 ordered pairs.
- Virtual route selection attempts are performed in the order in which the pairs are specified. The sequence of pairs, therefore, should reflect the desired route selection order. Refer to *OS/390 SecureWay Communications Server: SNA Customization* for information on the virtual route selection exit routine.

APPN Class-of-Service Definitions

Code an APPN class of service with the following definition statements:
- APPNCOS
- LINEROW
- NODEROW

Put the APPN COS definitions in one or more VTAMLST files that can be activated at VTAM initialization or with the VARY ACT command.

Table 69. APPN COS Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	APPNCOS	PRIORITY	674	R		MEDIUM	
		NUMBER	672			8	
	LINEROW	NUMBER	673	R			
		CAPACITY	670				
		COSTBYTE	671				
		COSTTIME	671				
		PDELAY	674				
		SECURITY	676				
		UPARM1	676				
		UPARM2	676				
		UPARM3	677				
		WEIGHT	677				
	NODE-ROW	NUMBER	673	R			
		CONGEST	670				
		ROUTERES	675				
		WEIGHT	678				

Legend:

R Required operand.

Coding Notes

APPNCOS Definition Statement: The APPNCOS definition statement begins the APPN COS definitions.

LINEROW Definition Statement: The LINEROW definition statement contains the operands which specify line characteristics.

NODEROW Definition Statement: The NODEROW definition statement contains the operands which specify node characteristics.

Implementing APPN Classes of Service

Details for implementing APPN classes of service are given in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

See “IBM-Supplied APPN Classes of Service” on page 680 for descriptions of the seven IBM-supplied APPN classes of service, and “APPN Class-of-Service (COS) Definitions (COSAPPN)” on page 798 for samples of them.

Full Syntax

The full syntax for the APPN COS follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

APPNCOS:

```
                                  ┌MEDIUM──┐   ┌NUMBER=8──────┐
►►──name──APPNCOS──PRIORITY=──┼────────┼───┼──────────────┼──────────────►◄
                                  ├HIGH────┤   │              │
                                  ├LOW─────┤   └NUMBER=──┬8──┬┘
                                  ├MEDIUM──┤             └12─┘
                                  └NETWORK─┘
```

LINEROW:

```
►►──LINEROW──NUMBER=1–8──┬──────────────────┬──┬──────────────────┬──►
                         └CAPACITY=min,max──┘  └COSTBYTE=min,max──┘

►──┬──────────────────┬──┬──────────────────┬──┬──────────────────┬──►
   └COSTTIME=min,max──┘  └PDELAY=min,max────┘  └SECURITY=min,max───┘

►──┬──────────────────┬──┬──────────────────┬──┬──────────────────┬──┬──────────────────┬──►◄
   └UPARM1=min,max────┘  └UPARM2=min,max────┘  └UPARM3=min,max────┘  └WEIGHT=0–255───────┘
```

NODEROW:

```
►►──NODEROW──NUMBER=1–8──┬──────────────────┬──┬──────────────────┬──►
                         └CONGEST=min,max───┘  └ROUTERES=min,max──┘

►──┬──────────────────┬──►◄
   └WEIGHT=0–255──────┘
```

Operand Descriptions

The following section describes the operands you can code for the APPN class of service.

CAPACITY:

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─CAPACITY=min,max─────┘
```

statements: LINEROW
dependencies: APPN only
range: 1K–1000M (for both *min* and *max*)

Specifies the range of the number of bits per second that the link can transmit data. The same range of values applies to *min* and *max*.

Table 70. Default Values for LINEROW CAPACITY Operand

Eight-Row Definitions		Twelve-Row Definitions	
The minimum default depends on the line row being defined:		The minimum default depends on the line row being defined:	
Row	**Minimum default**	**Row**	**Minimum default**
1	4M	**1**	100M
2	56000	**2**	25M
3	19200	**3**	25M
4	9600	**4**	1M
5	19200	**5**	64K
6	9600	**6**	28.8K
7	4800	**7**	100M
8	MINIMUM	**8**	25M
		9	64K
The maximum default is MAXIMUM for all line rows.		**10**	28.8K
		11	9.6K
		12	MINIMUM
		The maximum default is MAXIMUM for all line rows.	

CONGEST:

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─CONGEST=min,max──────┘
```

statements: *NODEROW*
dependencies: *APPN only*
range: *LOW–HIGH*

Indicates whether congestion is allowed for each node row.

CONGEST allows you to indicate the weighting preference for uncongested nodes, or to disallow the use of congested nodes.

A range of LOW,LOW indicates only uncongested nodes are allowed for the node row. A range of LOW,HIGH indicates uncongested and congested nodes are allowed for the node row.

min=LOW or HIGH and *max*=LOW or HIGH. The minimum default is LOW for all node rows. The maximum default depends on the node being defined:

Row	Maximum default
1	LOW
2	LOW
3	LOW
4	LOW
5	LOW
6	LOW
7	HIGH
8	HIGH

COSTBYTE:

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─COSTBYTE=min,max─────────────┘
```

statements: *LINEROW*
dependencies: *APPN only*
range: *0–255*

Specifies a range for cost-per-byte. The cost per byte is a relative value indicating the cost of the line per byte sent. Low values are less expensive than high values.

Table 71. Default Values for LINEROW COSTBYTE Operand

Eight-Row Definitions		Twelve-Row Definitions	
The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:		The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:	
Row	**Maximum default**	**Row**	**Maximum default**
1	0	**1**	0
2	0	**2**	0
3	0	**3**	0
4	0	**4**	0
5	0	**5**	0
6	128	**6**	0
7	196	**7**	128
8	255	**8**	128
		9	128
		10	128
		11	0
		12	255

COSTTIME:

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─COSTTIME=min,max─────────────┘
```

statements: LINEROW
dependencies: APPN only

range: 0–255

Specifies a range for cost-per-connect-time. The cost per connect time is a relative value indicating the cost of the line per unit of connect time. Low values are less expensive than high values.

Table 72. Default Values for LINEROW COSTTIME Operand

Eight-Row Definitions		Twelve-Row Definitions	
The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:		The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:	
Row	**Maximum default**	**Row**	**Maximum default**
1	0	**1**	0
2	0	**2**	0
3	0	**3**	64
4	0	**4**	64
5	0	**5**	64
6	128	**6**	64
7	196	**7**	64
8	255	**8**	64
		9	128
		10	128
		11	64
		12	255

name:

```
►►──name──────────────────────────────────────────────────────────►◄
```

statements: *APPNCOS*
dependencies: *APPN only*

Specifies a required 1–8 character name for the APPNCOS definition statement. The name must be unique within a node.

Note: If you reuse the name of an APPN class-of-service definition, the operand values of the later definition replace the operand values of the earlier definition.

NUMBER (APPNCOS):

```
                ┌─NUMBER=8──────────┐
►►──────────────┤                   ├──────────────────────────────►◄
                │           ┌─8─┐   │
                └─NUMBER=───┤   ├───┘
                            └─12─┘
```

statements: *APPNCOS*
dependencies: *APPN only*

Specifies which set of default values is to be used for the APPN COS table. If you specify 8, the default values for 8-row definitions are in effect. If you specify 12, the default values for 12-row definitions are in effect.

See “IBM-Supplied APPN Classes of Service” on page 680 for information on how to determine which set of default values is appropriate for the COS table you are defining.

NUMBER (LINEROW):

```
►►──NUMBER=1–12──────────────────────────────────────────────────────────►◄
```

statements:	*LINEROW*
dependencies:	*APPN only*

Specifies a line row to be defined.

The default characteristic values of a given line row can differ depending on the number of the row being defined. So, an operand might have a different value for line row 7 than for line row 8. See the line row operand descriptions for defaults for each row number.

Line rows are searched in ascending order to determine whether a given TG is acceptable for a particular COS. For example, when evaluating TG1 for use in route selection, its characteristics are compared against line row 1 for COSX. If TG 1’s characteristic values are within the acceptable range of values for the characteristics of line row 1, TG 1 is assigned the weight specified for line row 1. If TG 1’s characteristics are not within the acceptable range, the next defined line row for COSX is compared against TG 1. If no COSX line rows are found that agree with TG 1’s characteristic values, TG 1 is assigned an infinite weight and is not used to calculate a route for COSX.

A line row is defined when its number is specified, even though no other line row operands are coded.

NUMBER (NODEROW):

```
►►──NUMBER=1–8───────────────────────────────────────────────────────────►◄
```

statements:	*NODEROW*
dependencies:	*APPN only*

Specifies a node row to be defined.

The default characteristic values of a given node row can differ depending on the number of the row being defined. So, an operand might have a different value for node row 7 than for node row 8. See the node row operand descriptions for defaults for each row number.

Node rows are searched sequentially based on this numbering to determine whether a given node is acceptable for a particular COS.

For example, when NODE A becomes active, its characteristics are compared against node row 1 for COSX. If NODE A’s characteristic values are within the acceptable range of values for the characteristics of node row 1, then NODE A is assigned the weight specified for node row 1. If NODE A’s characteristics are not within the acceptable range, the next defined node row for COSX is compared

against NODE A. If no COSX node rows are found that meet NODE A's characteristic values, NODE A is assigned an infinite weight and is not used for route calculation for COSX.

A node row is defined when its number is specified, even though no other node row operand are specified.

PDELAY:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └PDELAY=min,max────┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*see values*

Specifies a range for propagation delay. Propagation delay represents the time needed for a signal to travel from one end of the link to the other.

min and *max* have the same range of the following values:

MINIMUM
Indicates minimum delay.

NEGLIGIB
Indicates a local area network delay (less than 0.48 millisecs).

TERRESTR
Indicates telephone network delay (between 0.48 and 49.152 millisecs).

PACKET
Indicates a packet-switched network delay (between 49.152 and 245.76 millisecs).

LONG
Indicates a satellite delay (greater than 245.76 millisecs).

MAXIMUM
Indicates a maximum delay.

Table 73. Default Values for LINEROW PDELAY Operand

Eight-Row Definitions	Twelve-Row Definitions
The minimum default is MINIMUM for all line rows. The maximum default depends on the line row being defined:	The minimum default is MINIMUM for all line rows. The maximum default is MAXIMUM for all line rows.
Row **Maximum default** **1** NEGLIGIB **2** TERRESTR **3** TERRESTR **4** TERRESTR **5** PACKET **6** PACKET **7** MAXIMUM **8** MAXIMUM	

PRIORITY:

```
                  ┌─MEDIUM──┐
►►──PRIORITY=─┬──────────┬──────────────────────────────────────────►◄
              │          │
              ├─HIGH────┤
              ├─LOW─────┤
              ├─MEDIUM──┤
              └─NETWORK─┘
```

statements: *APPNCOS*
dependencies: *APPN only*

Specifies one of the following transmission priorities for an APPN COS:

PRIORITY=LOW
Recommended for batch traffic

PRIORITY=MEDIUM
Recommended for normal traffic

PRIORITY=HIGH
Recommended for interactive traffic

PRIORITY=NETWORK
Used for APPN network services traffic, and is valid only for CPSVCMG and SNASVCMG.

ROUTERES:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─ROUTERES=min,max─┘
```

statements: *NODEROW*
dependencies: *APPN only*
range: *0–255*

Specifies the range of the relative amount of route addition resistance (RAR) for each node row. The route addition resistance is used to identify network nodes that are more preferable for intermediate routing compared to other network nodes. A low RAR value indicates a more desirable node for intermediate traffic.

The minimum default is 0 for all node rows. The maximum default depends on the node row being defined:

Row	Maximum default
1	31
2	63
3	95
4	127
5	159
6	191
7	223
8	255

SECURITY:

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─SECURITY=min,max─────────┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*see values*

Specifies a range from the minimum security level to the maximum security level.

min and *max* can each be one of the following values, which are listed from least to most secure:

UNSECURE
Indicates not secured.

PUBLIC
Indicates public switched network.

UNDERGRO
Indicates unguarded underground cable.

SECURE
Indicates unguarded secure conduit.

GUARDED
Indicates guarded conduit, physical only.

ENCRYPT
Indicates link encryption.

SHIELDED
Indicates guarded conduit, physical and radiation shielded.

MAXIMUM
Indicates maximum security (equivalent to *SHIELDED*).

The minimum default is UNSECURE for all line rows. The maximum default is MAXIMUM for all line rows.

UPARM1:

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─UPARM1=min,max───────────┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*0–255*

Specifies a range for a user-defined value. The user determines the meaning of this value.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for an example of using the UPARM operands.

UPARM2:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─UPARM2=min,max───┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*0–255*

Specifies a range for a user-defined value. The user determines the meaning of this value.

UPARM3:

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─UPARM3=min,max───┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*0–255*

Specifies a range for a user-defined value. The user determines the meaning of this value.

WEIGHT (LINEROW):

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └─WEIGHT=0–255─────┘
```

statements:	*LINEROW*
dependencies:	*APPN only*
range:	*0–255*

Specifies the relative weight for each line row. Low weights indicate more desirable choices.

Note: An ascending order of weights on a row-by-row basis will not be enforced. For example, line row 1 can be assigned a weight that is greater than line row 2. However, this is not recommended.

Table 74. Default Values for LINEROW WEIGHT Operand

Eight-Row Definitions		Twelve-Row Definitions	
The default depends on the line row being defined:		The default depends on the line row being defined:	
Row	**Default**	**Row**	**Default**
1	30	**1**	20
2	60	**2**	40
3	90	**3**	60
4	120	**4**	80
5	150	**5**	100
6	180	**6**	120
7	210	**7**	140
8	240	**8**	160
		9	180
		10	200
		11	220
		12	240

WEIGHT (NODEROW):

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    └─WEIGHT=0–255─┘
```

statements: *NODEROW*
dependencies: *APPN only*
range: *0–255*

Specifies the relative weight for each node row. Low weights indicate more desirable choices.

Note: An ascending order of weights on a row-by-row basis is not enforced. For example, node row 1 can be assigned a weight that is greater than node row 2. However, this is not recommended.

The default depends on the node row being defined:

Row	Default
1	5
2	10
3	20
4	40
5	80
6	100
7	120
8	150

Default Values for APPN Class-of-Service Definitions

You can use default values for APPN COS operands. The examples in this section illustrate the use of defaults for APPN class-of-service definitions along with descriptions of how default values are substituted.

Note: Defaults apply only to partially coded rows. If a row is not coded, no defaults are assumed.

Example 1: WEIGHT Operand not Coded: In the following example, the WEIGHT operand was not specified. Because the line row number was specified as 5 and the default values for 8-row definitions are in effect (NUMBER=8 on the APPNCOS definition statement), the value 150 is substituted as the default value for WEIGHT. Had the default values for 12-row definitions been in effect (NUMBER=12 on the APPNCOS definition statement), the value 100 would have been substituted as the default value for weight. Note also that line rows 1, 2, 3, 4, 6, 7, and 8 were not coded. Therefore, only one line row is defined for COSDEF1.

```
************************************************************************
* COS entry                                                            *
************************************************************************
COSDEF1  APPNCOS PRIORITY=HIGH                  transmission priority
************************************************************************
* Line row                                                             *
************************************************************************
         LINEROW NUMBER=5,                      line row number       *
                 CAPACITY=(1200,4M),            line speed            *
                 COSTTIME=(5,10),               cost per connect time *
                 COSTBYTE=(10,20),              cost per byte         *
                 PDELAY=(MINIMUM,PACKET),       propagation delay     *
                 SECURITY=(PUBLIC,ENCRYPT),     security              *
                 UPARM1=(0,100),                user defined parm 1   *
                 UPARM2=(100,200),              user defined parm 2   *
                 UPARM3=(0,10)                  user defined parm 3
```

Example 2: Minimum Value for COSTTIME Operand not Coded: In the following example, the minimum value for COSTTIME was not specified. Because the line row number was specified as 2, the value 0 is substituted as the minimum default value for COSTTIME. In this case, the default value for 8-row definitions is being used because NUMBER is not coded on the APPNCOS definition statement and it therefore defaults to 8. However, even if NUMBER=12 was coded on the APPNCOS definition statement, the value 0 would still be substituted as the minimum default value for COSTTIME because for line row 2, the minimum default value is 0 for both 8-row and 12-row definitions. Also, UPARM1 was not coded. The values 0 and 255 are substituted for the UPARM1 minimum and maximum values respectively.

```
************************************************************************
* COS entry                                                            *
************************************************************************
COSDEF2  APPNCOS PRIORITY=HIGH                 transmission priority
************************************************************************
* Line row                                                             *
************************************************************************
         LINEROW NUMBER=2,                      line row number       *
                 WEIGHT=20,                     line row weight       *
                 CAPACITY=(1200,4M),            line speed            *
                 COSTTIME=(,10),                cost per connect time *
                 COSTBYTE=(10,20),              cost per byte         *
                 PDELAY=(MINIMUM,PACKET),       propagation delay     *
```

```
                SECURITY=(PUBLIC,ENCRYPT),      security              *
                UPARM2=(100,200),               user defined parm 2   *
                UPARM3=(0,10)                   user defined parm 3
```

Example 3: One Line Row Defined for COSDEF3: In the following example, one line row has been defined for COSDEF3. Because the only operand specified for the line row was its number, default values are substituted for the remaining line row characteristics. These values are obtained from the appropriate set of defaults for line row 8, which in this case, is the set of defaults for 8-row definitions (because NUMBER is not coded on the APPNCOS definition statement and it therefore defaults to 8). Had NUMBER=12 been coded on the APPNCOS definition statement, the set of defaults for 12-row definitions would have been substituted for the remaining line row characteristics for line row 8. The same situation applies for the node row coded in this example, except that the default values are obtained from the set of defaults for node row 1.

```
************************************************************************
* COS entry                                                            *
************************************************************************
COSDEF3  APPNCOS PRIORITY=HIGH                  transmission priority
************************************************************************
* Line row                                                             *
************************************************************************
         LINEROW NUMBER=8                       line row number
************************************************************************
* Node row                                                             *
************************************************************************
         NODEROW NUMBER=1                       node row number
```

IBM-Supplied APPN Classes of Service

Descriptions of the seven IBM-supplied APPN classes of service follow. These classes define seven generic types of session traffic. See “APPN Class-of-Service (COS) Definitions (COSAPPN)” on page 798 for samples.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more details.

The IBM-supplied classes of service are shipped with the name COSAPPN in SYS1.ASAMPLIB and should be copied into SYS1.VTAMLST at VTAM installation.

Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for more details.

#BATCH
An APPN COS for LU-LU sessions that specifies a general batch-oriented COS that uses low transmission priority, and for which high bandwidth and low cost are considered more important than short delay.

#BATCHSC
An APPN COS for LU-LU sessions that specifies a general batch-oriented COS that uses low transmission priority, and for which high bandwidth and low cost are considered more important than short delay. A minimum security level is required.

#CONNECT
An APPN COS for LU-LU sessions that provides connectivity at medium transmission priority.

#CONNECT should not be changed.

CPSVCMG
An APPN COS for CP-CP sessions that is used for network flows. It provides connectivity at network transmission priority.

CPSVCMG should not be changed.

#INTER
An APPN COS for LU-LU sessions that specifies a general, interactive class of service that uses high transmission priority, and for which short delay is considered more important than high bandwidth and lost cost.

#INTERSC
An APPN COS for a LU-LU session that specifies a general, interactive COS that uses high transmission priority, and for which short delay is considered more important than high bandwidth and lost cost. A minimal security level is required.

SNASVCMG
An APPN COS for LU-LU CNOS sessions that provides connectivity at network transmission priority.

SNASVCMG should not be changed.

CMIP Services Directory Definition File

If you specified the OSIMGMT=NO start option, you do not need to read this section.

If you specified the OSIMGMT=YES start option, a directory definition file is required. This section describes how to code the directory definition file to implement Common Management Information Protocol (CMIP) services.

The CMIP services directory definition file performs name mapping and address mapping, and controls the associations that CMIP services on this host has with other CMIP services and with itself. An association is a connection between CMIP services on this host and CMIP services on another node or between CMIP services on this host and itself.

The IBM-supplied directory definition file in member ACYDDF of the SYS1.SISTCMIP data set allows unlimited associations with other instances of CMIP services and between CMIP services and itself. If you do not update the IBM-supplied file, any CMIP application program in any network can have access to CMIP services on this host.

To learn about the purpose of the CMIP services directory definition file, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

This section covers the following topics:
- Example of a simple directory definition file
- Explanation of the example directory definition file
- Specifying different types of associations
- Coding the directory definition file
- How CMIP services uses the directory definition file
- Troubleshooting the directory definition file

Example of a Simple Directory Definition File

The following example shows a simple directory definition file for NETA. An explanation follows the example.

```
# Example directory definition file.

class aetitle

class aetitle
    name '1.3.18.0.2.4.6=NETA'
    associationKey '*'

class aetitle
    name '1.3.18.0.2.4.6=NETB'
    associationKey 'a0b1c2d3e4f50011'

class aetitle
    name '*'
    associationKey '-'
```

Explanation of the Example Directory Definition File

The directory definition file contains comments and directory entries:

- Comments are not processed. They begin with the # symbol in column 1. Blank lines are treated as comments.
- Directory entries determine which associations are allowed, and for those associations, what type of security is to be used.

Each directory entry contains attributes and their values. An attribute is similar to a VTAM operand, in that the attribute and the value specified on it define a characteristic of an object. In the directory definition file, the object is CMIP services.

The value is a single expression, which can contain blanks if the value is enclosed in quotation marks. Everything after the value in a directory entry record is treated as a comment record and is not processed.

An entry ends where the next entry begins (with a different name or class attribute) or at the end of the directory definition file.

The first attribute in the example directory definition file is the class attribute, with the value aetitle. See “Format of Directory Definition File Entry Records” on page 684 for an explanation of class. This entry specifies that the names under the class attribute are assumed to be CMIP services names or names of CMIP application programs that have registered their own AE-titles. For a description of AE-titles, refer to *OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide*.

The class and its attribute are followed by the second attribute, name with a value. The required name attribute is the first attribute for an object that belongs in the class specified by the preceding class attribute. CMIP services searches for the name attribute in the directory definition file when it determines whether a particular instance of CMIP services can have an association with CMIP services on this host.

The third attribute is the associationKey attribute with a value. Specifying associations is discussed in the following section.

Specifying Different Types of Associations

In the directory definition file, you can specify the following options for establishing associations:

- No security on associations
- Data encryption standard (DES)-based security on associations
- Application-program-to-application-program security on associations
- No associations

No Security on Associations: The following entry

```
class aetitle
    name '1.3.18.0.2.4.6=NETA'
    associationKey '*'
```

specifies that all instances of CMIP services on network NETA can exchange requests and actions with the CMIP services on this host. All associations are allowed.

DES-Based Security on Associations: The following entry

```
class aetitle
    name '1.3.18.0.2.4.6=NETB'
    associationKey 'a0b1c2d3e4f50011'
```

specifies that some instances of CMIP services on NETB can exchange requests and actions with CMIP services on this host. Associations are allowed only if the correct DES encryption key is used. Note that the directory definition file on any instance of CMIP services on NETB is required to have the same DES encryption key for the association to be allowed.

In the following example, the combination of the * value for the name attribute and the associationKey attribute establishes an encryption key that must be used by all instances of CMIP services that are not already specified with a name attribute on another entry in the directory definition file.

```
class aetitle
    name '*'
    associationKey '0123456789abcdef'
```

This entry specifies that all associations with CMIP services on this host must use the same encryption key, `0123456789abcdef`, to have an association with CMIP services on this host.

Application-Program-to-Application-Program Security on Associations: The following entry

```
class aetitle
  name '1.3.18.0.2.4.6=NETA'
  associationKey '.'
```

specifies that the CMIP application programs control access to themselves, rather than CMIP services controlling access to the application programs.

If you want to use security, but do not want DES-based security, the CMIP application programs can define application-program-to-application-program security. The VTAM topology agent and the NetView program do not implement application-program-to-application-program security. For information about defining this type of security, refer to the *OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide*.

No Associations: The following entry

```
class aetitle
    name '*'
    associationKey '-'
```

specifies that no associations are allowed for instances of CMIP services on other networks, except NETA and NETB, since they are specifically listed with a name attribute and an associationKey attribute. The asterisk specified as the value of the name attribute means that the directory entry applies to any name not specifically listed in another directory entry in the file.

Coding the Directory Definition File

This section explains the format and rules used to code entries in the directory definition file.

Format of Directory Definition File Entry Records: Directory entry records can have one or more blanks (for indentation) before the keyword. There can be multiple blanks between the keyword and the value. The value is a single string and can contain blanks only when it is enclosed in quotes. Everything after the value in a record is treated as a comment and is ignored.

A minimum entry consists of a name line and one or more attributes associated with that name. Each entry is ended by the next entry (class or name) or the end of the file. More specifically, an entry consists of the following:

- The class of a list of entries. The class applies to all succeeding entries until another class is specified. The first non-blank item on the line is the keyword “class”, followed by the class attribute, or type (separated from the “class” keyword by at least one blank). The only types of classes are:
  - managed object
  - aetitle

  The first entry in the file must start with a class record, or a “no class specified” message is issued, and initialization or reloading of the table fails. Subsequent entries can omit the class record if these entries belong to the last class specified.
- The name attribute. This is the string against which the search is made, within the class. The first non-blank item on the line is the keyword “name”, followed by the value of the name (a string in single quotes), separated from the keyword by at least one blank. Multiple entries with the same name value may exist in the directory definition file, providing all attributes under entries with that name value are different.
- Following the name attribute, an attribute associated with an entry. The first non-blank item on the line is the keyword for a particular attribute, followed by the value (a string in single quotes) of the attribute, separated from the keyword by at least one blank. The current attributes used by CMIP services for the class, managed object, are:
  - name
  - aetitle

  The current attributes used by CMIP services for the class, aetitle, are:
  - name
  - address
  - associationKey
  - timeSync

If an attribute is encountered that has already been defined for a particular name value (for example, a duplicate attribute entry, such as two different associationKey values), the second one produces a “duplicate attribute” message, and initialization or table reloading fails.

Individual attributes are optional on any particular entry.

Each record in the file can extend over many lines, up to a maximum of 2080 character spaces. This includes any leading spaces before an attribute at the beginning of the line. The maximum logical record length (LRECL) for the directory definition file is 2080. The record format (RECFM) for the file must be FIXED.

Code entries as one or more records in the following format:

Column 1	**End of line**
[...]attribute[value]...	**...&**

... Indicates blanks. One or more blanks can precede the first attribute.

attribute
Represents a characteristic of the managed object. Directory entry records can have one or more blanks before the attribute and can have multiple blanks between the attribute and the value. Separate the attribute from the value with one or more blanks.

value The value can have intervening blanks, if it is enclosed in quotation marks.

& If the entry requires several records, code the continuation character (&) in the last position on the line.

Comment lines begin with the # symbol in column 1. Comments are not processed. Blank lines are treated as comments.

Following is a sample entry in a directory definition file. The first line is a scale to show column positions. Every tenth column position is indicated by a number.

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...
class managed object
   name     '1.3.18.0.2.4.6=USIBMNR;2.9.3.2.7.4=(name NR55550I)'
   aetitle  '1.3.18.0.2.4.6=USIBMNR;2.9.3.2.7.4=(name "NR55555I");1.3.18.0.2.4.12=OSISMASE'
```

Observe the following rules if more than one record is required for an entry.

- Code the line up to the point where you want to break the line. This point can be anywhere on the line.
- At the end of the line, code the continuation character &; The & symbol does not form part of the line that is processed.
- Continue the entry in the first column of the next line. A comment line cannot follow a continuation line.
- Continue coding lines with continuation characters until the entry is finished.
- Entries can be coded on as many lines as needed.

Attributes Coded in the Directory Definition File: For VTAM CMIP services, the following attributes are valid:

- On the class, managed object
  - name
  - aetitle
- On the class, aetitle

- name
- address
- associationKey
- timeSync

How to Code the name Attribute: The first non-blank item on the line is the name attribute followed by the value of the name attribute, which is enclosed in single quotation marks.

At least one blank must separate the name attribute from the value for the name attribute. The value of the name attribute is an AE-title.

The format of the value for the name attribute can vary. The actual value to be used is determined by the manager or agent application program.

Multiple entries with the same value for the name attribute can exist in the directory definition file when all attributes under entries with that name value are different.

The following examples show how to specify the value for the name attribute. The value stands for an instance of CMIP services.

- The name of CMIP services on a particular network-qualified CP or SSCP name. An ampersand causes the line to continue.

  ```
  name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSCP1A);&
  1.3.18.0.2.4.12=OSISMASE'
  ```

  The name consists of object identifiers and values. Object identifier 1.3.18.0.2.4.6 means the SNA network node name:
  - ISO has assigned 1.3.18 to IBM.
  - IBM has assigned 0.2.4.6 to mean a SNA network node name.

  The other object identifiers are similarly assigned:
  - Object identifier 2.9.3.2.7.4 means a SNA node name, either a CP or an SSCP name.
  - Object identifier 1.3.18.0.2.4.12 means the application entity title (AE-title). Generally, the AE-title refers to CMIP services, which is always OSISMASE for VTAM.
- A network-qualified CP or SSCP name by itself:

  ```
  name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSCP1A)'
  ```

  A trailing wildcard character ('*') can be used in the CP or SSCP name to indicate that the rule matches any network-qualified CP or SSCP name with the specified netid and a CP or SSCP name that matches up through the wildcard character and has zero or more characters after it. If the wild card is specified, it must be the last character in the CP or SSCP name.

  For example, the following name attribute rules match NETA.SSCP2B:

  1.
     ```
     name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name S*)'
     ```
  2.
     ```
     name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSSC*)'
     ```
  3.
     ```
     name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSSCP2*)'
     ```
  4.

```
name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSSCP2B*)'
```

5.

```
name '1.3.18.0.2.4.6=NETA;2.9.3.2.7.4=(name SSSCP2B)'
```

If an exact match is specified, such as 687 in the previous example, it will be selected, even if there are one or more wildcard rules which match, such as 686, 686, 686, and 687 in the previous example.

If more than one exact match is specified, it is indeterminate which exact match will be selected by CMIP Services.

If no exact match is specified and multiple wildcard matches are specified, it is indeterminate which wildcard match will be selected by CMIP Services.

- A network name by itself:

  ```
  name '1.3.18.0.2.4.6=NETA'
  ```

- Global match, which matches any name:

  ```
  name '*'
  ```

How to Code the address Attribute: Values for address records are specified according to the protocol used to communicate over the association.

You can specify a network ID and node in the address attribute, rather than allowing CMIP services to extract them from the AE-title.

How to Code the aetitle Attribute: The aetitle attribute is composed of three parts:
- The network ID
- The node or network addressable unit
- The System Management Application Entity (SMAE) name

The three parts are separated by semicolons. Except within the node part, there are no embedded blanks.

```
class managed object
   name     '1.3.18.0.2.4.6=USIBMNR;2.9.3.2.7.4=(name NR55550I)'
   aetitle  '1.3.18.0.2.4.6=USIBMNR;2.9.3.2.7.4=(name "NR55555I");1.3.18.0.2.4.12=OSISMASE'
```

- The object identifier 1.3.18.0.2.4.6 is assigned the value equivalent to the network ID for the first part.
- The second part consists of the node ID (CP Name) associated with object identifier 2.9.3.2.7.4, in the form (name ″nodeID″). Note that the above example shows the nodeID both by itself and surrounded by double-quotes (″). Algorithmic mapping performed by directory services always generates an AE-title with the node ID enclosed in double-quotes. Therefore, they should also be specified in any directory definition file “aetitle” records. However, directory services allows a match even when the directory definition file record does not have the node ID enclosed in quotes.
- The third part is the SMAE name, “OSISMASE” in the above example. It is the value for object identifier 1.3.18.0.2.4.12.

In general, when interconnecting management networks, the network administrator or other persons updating the directory definition file must determine how the aetitle is to be specified for each remote partner for which an entry must be made.

How to Code the associationKey Attribute: The associationKey attribute specifies one of the following options for the association:
- No security

- DES-based security
- Application-program-to-application-program security
- No association

The value can be a hexadecimal representation of the actual DES encryption key. Hexadecimal representation is used to avoid problems between systems using EBCDIC and systems using ASCII for character representations.

Unless the key is one of the three special character values described in the following list, the key must be 16 hexadecimal digits long. This is essentially the same as having a 16-character key consisting of digits 0-9 and letters A-F. Either uppercase or lowercase letters can be used, though VTAM translates and processes letters in uppercase.

The four associationKey values specify how access is to be controlled. To indicate that:

- All instances of CMIP services have access to CMIP Services on this host, specify:

  ```
  associationKey '*'
  ```

- CMIP services controls access to itself by using DES encryption, specify the value of the hexadecimal key used for DES encryption:

  ```
  associationKey 'a0b1c2d3e4f50011'
  ```

- CMIP applications programs (both manager and agent application programs) control access to themselves, rather than CMIP services controlling access to the application programs, specify:

  ```
  associationKey '.'
  ```

  Note: The VTAM topology agent and the NetView program do not implement application-program-to-application-program security.

- No associations are allowed, specify:

  ```
  associationKey '-'
  ```

How to Code the timeSync Attribute: The timeSync attribute specifies the maximum time difference (specified in seconds) that is allowed between the current time of day (specified in date, hours, minutes, and seconds) on this host and the current time of day on the node of the partner CMIP services.

The value is an absolute value. For example, if you specify 4 seconds as the value for the timeSync parameter, the system time of day can be either 4 seconds ahead of or 4 seconds behind the value received.

The default is 300 seconds (5 minutes). The maximum value allowed is 86400 (1 day).

If a timeSync value of 0 is specified, time-of-day checking is disabled.

How CMIP Services Uses the Directory Definition File

The directory definition file contains names of other instances of CMIP services and of itself. When searching the directory definition file, CMIP services uses the most specific name it can find in the directory definition file. CMIP services searches the directory definition file for the attribute value for a particular name. Here is an example of a directory definition file, followed by some sample searches. (The line numbers are shown for reference only; they cannot appear in a valid directory definition file.)

```
 1 class aetitle
 2   name '*'
 3     timeSync 800
 4
 5 class aetitle
 6   name '1.3.18.0.2.4.6=SNANET1'
 7     associationKey 'd6c8f0d3e4F7c5c2'
 8     timeSync 17500
 9
10 class aetitle
11   name '1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED)'
12     address USIBMNR.RALVMS
13     associationKey '*'
14
15 class aetitle
16   name '1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED);&
17 1.3.18.0.2.4.12=BadSMASE'
18     associationKey '-'
```

CMIP services searches the directory definition file for a particular attribute for a given full name. Here are some sample searches:

- Search for the associationKey attribute for name

  `1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED);14 1.3.18.0.2.4.12=BadSMASE'`

  In this example, line 18 is used (name matches exactly and associationKey exists for that specific name).
- Search for the associationKey attribute for name

  `1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED);14 1.3.18.0.2.4.12=OSISMASE'`

  In this example, line 13 is used. The exact name is not in the directory definition file, so CMIP services searched for name ’1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED)’ in the directory definition file and found it, and it had an associationKey.
- Search for the associationKey attribute for name

  `1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name UNSECURED);14 1.3.18.0.2.4.12=OSISMASE'`

  In this example, line 7 is used. The exact name, and the first 2/3 name are not found, but name ’1.3.18.0.2.4.6=SNANET1’ is found, and it has an associationKey.
- Search for the associationKey attribute for name

  `1.3.18.0.2.4.6=OTHERNET;2.9.3.2.7.4=(name GEORGE);14 1.3.18.0.2.4.12=OSISMASE'`

  In this example, name is not found, which is equivalent to associationKey ’*’ (all associations allowed).
- Search for the timeSync attribute for name

  `1.3.18.0.2.4.6=SNANET1;2.9.3.2.7.4=(name SECURED);14 1.3.18.0.2.4.12=OSISMASE'`

  In this example, line 8 is used. Note that line 11 matches this name (the first 2/3 of this name, actually), but there is no timeSync for that name. So CMIP services keeps searching, and finds a match for the first 1/3 of the name and it does have a timeSync value.

Troubleshooting the Directory Definition File

If you receive the message IST1444I *filename* NOT LOADED – DUPLICATE ATTRIBUTE AT RECORD *record*, CMIP services has encountered an attribute that has already been defined for an object. The directory definition file fails to load. For example, if

the associationKey attribute has already been defined with a value and a different associationKey value is encountered for the same value on the name attribute, message IST1444I is issued.

If you receive the message IST1444I *filename* NOT LOADED - SYNTAX ERROR AT RECORD *record*, CMIP services has encountered an attribute that is not valid. The directory definition file fails to load.

If you receive the message IST1444I *filename* NOT LOADED - NO CLASS SPECIFIED, the class attribute is missing. The directory definition file fails to load.

For other examples, refer to the description of message IST1444I in *OS/390 SecureWay Communications Server: SNA Messages*.

Interpret Table

When VTAM receives a logon or logoff request, it uses the interpret table to determine which application program is to be notified.

The standard logon procedure should meet the needs of most installations. (See “LOGON Command” on page 758 for the standard logon procedure.) But, you can write your own interpret table for special circumstances (for example, if the logon sequence you want to use does not follow the syntax for USS commands).

IBM does not supply a default logon-interpret routine, but if you need one, you can code it.

Note: An independent logical unit does not support “uninterpreted” names; therefore, the interpret table does not apply to names received from an independent logical unit.

Table 75. Interpret Table Macroinstructions and Operands

Name	Macro-instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	INTAB					
name	LOGCHAR	APPLID	695	R		
		REMOVE	696		N	
		SEQNCE	697			
name	ENDINTAB					

Legend:

R Required operand.

Coding Notes

INTAB Macroinstruction

The INTAB macroinstruction defines an interpret table that lists the VTAM application programs with which one or more logical units can establish a session. One INTAB macroinstruction defines the name of the interpret table and a group of logon message definitions.

LOGCHAR Macroinstruction

Each LOGCHAR (logon-characters) macroinstruction defines a single logon message and the name of an application program, a logon-interpret routine, or a USERVAR. You can include more than one LOGCHAR macroinstruction in an interpret table.

VTAM compares the logon message (character by character) with successive entries in the specified interpret table. If the leading characters in the logon message correspond to all the characters in an entry in the interpret table, VTAM accepts the logon message as valid (even though the logon message can be longer than the corresponding entry in the interpret table). If the first character or characters of several logon messages are identical, you should arrange the LOGCHAR macroinstructions so the logon sequences for the logon messages are from the most restrictive (greatest number of characters) to the least restrictive (fewest number of characters). For example:

```
SEQ1 LOGCHAR APPLID=(APPLICID,AP2),SEQNCE='LOG2'
SEQ2 LOGCHAR APPLID=(APPLICID,AP1),SEQNCE='LOG'
```

Otherwise, in the preceding example, if sequence LOG had preceded LOG2 in the interpret table, both logon messages LOG and LOG2 would be valid logons to application program AP1. If you use two or more LOGCHAR macroinstructions, they must be arranged so that their SEQNCE fields are in reverse collating order.

Note: Although you can define a USERVAR in the interpret table, you can use the interpret table entry in conjunction with a USS table to process an incoming character-coded LOGON. For details, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

ENDINTAB Macroinstruction

The ENDINTAB macroinstruction defines the end of an interpret table. Code one ENDINTAB macroinstruction after one or more LOGCHAR macroinstructions to define the end of an interpret table. You can also follow the ENDINTAB macroinstruction with an assembler language END statement or with CSECTs containing one or more user-written logon-interpret routines.

If you code an assembler language END statement, it must be in the format:

```
END name
```

where *name* is the label of the INTAB macroinstruction and specifies the main entry point.

Follow the ENDINTAB macroinstruction with an assembler language END statement unless the interpret table is to be followed by CSECTs containing one or more user-written APPLID routines, as described in “Coding Logon-Interpret Routines” on page 692.

Coding Logon-Interpret Routines

You can code logon-interpret routines to validate logons and determine the name of the application program that is to receive the logons. The entry point name must match the *routine name* specified in the APPLID=(ROUTINE,*routine name*) operand in the LOGCHAR macroinstruction. All logon-interpret routines specified in an interpret table must be assembled and link-edited with that interpret table.

The logon-interpret routine interface allows the routine to supply a network-qualified application name for interpreted logons.

If you want the logon-interpret routine to supply a network-qualified application name, you need to change the interpret routine parameter list. If you do not want the routine to supply a network-qualified name, you do not need to change the routine parameter list. Refer to *OS/390 SecureWay Communications Server: SNA Customization* for details on coding the interpret routine parameter list.

Requirements for Logon-Interpret Routines

Entry from:
VTAM

Entry point:
routine name

Contents of registers at entry:
Register 0:
Length of logon message
Register 1:
Address of first byte of logon message. For LOGON requests, VTAM searches the interpret table again, after USS translation, looking only for the specified APPLID. After USS translation, register 1 contains the address of the first byte of the APPLID.
Register 2:
Address of an 8-byte logical unit name
Register 4:
Address of parameter list for the network identifier and resource name.
Register 13:
Address of a 72-byte save area provided by VTAM
Register 14:
Return address
Register 15:
Address of entry point of this routine.

Operation: The logon-interpret routine is run synchronously in pageable storage under the control of VTAM and not under the control of an application program. For the application program to receive the logon, this routine must validate the logon, obtain the name of the application program to receive control, and provide this name to VTAM. Otherwise, the routine specifies that the logon is not valid or that the name of the application program was not found.

The logon-interpret routine must also:

- Save and restore the contents of registers 2–14 when receiving and passing control
- Use re-enterable code (the routine must not store anything within itself or modify itself during execution)

- Perform no I/O operations; an I/O request causes the routine to terminate abnormally.

Because the logon-interpret routine operates at VTAM's main task dispatching priority, there is a possibility of lockout if a wait requires another task action. The routine gets control in supervisor state with a VTAM storage key, so errors within the routine could cause damage to VTAM or to system control blocks and modules.

Notes:

1. You can modify the logon message that is passed to the interpret routine. However, remember these two points:
   - VTAM does not look at the changed storage; it is passed as user data to the application.
   - You should modify with caution, as modification outside the message storage boundaries could result in VTAM outages.
2. All data is addressable only in 24-bit mode.

Contents of Registers at Exit: Registers 0 and 1 contain the name of the application program (in EBCDIC characters) with which the logical unit is to establish a session:

Register 0:
First 4 characters of name (left-justified).

Register 1:
Last 4 characters of name (left-justified).

Registers 2–14:
Restored to condition at entry.

Register 15:
Return code:

00 Application program was found and the name is placed in registers 0 and 1.

Nonzero
Application program was not found and the name is not placed in registers 0 and 1.

If the name of the application program contains fewer than 8 characters, use blanks to provide a name with 8 characters.

Installing and Changing Interpret Tables

Follow these steps to install each interpret table and any user-written routines:

1. Assemble the interpret table and the user-written routines referred to by the LOGCHAR macroinstruction.

   Put the user-written routines in a private call library.
2. Link-edit the interpret table with its associated routines, preferably assigning a module name that matches the name of the interpret table as specified with the INTAB macroinstruction.

   Link-edit the resulting module as a non-executable module into a library identified by the SYS1.VTAMLIB DD statement.
3. Code the assigned name on the LOGTAB operand of the appropriate definition statement to associate the logical unit or logical units with this interpret table

(refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on specifying the LOGTAB operand in definition statements).

To change an installed interpret table by adding a LOGCHAR macroinstruction, or to replace an installed interpret table, follow these steps:

1. Add or replace the LOGCHAR macroinstruction in the source program.
2. Assemble and link-edit the new interpret table, thus replacing the old module.

The control sections of the interpret table module consist of the interpret table itself and a CSECT for each user-written routine identified by the ROUTINE operand of the LOGCHAR macroinstruction. Individual CSECTs can be extracted when the interpret table is link-edited. Because reprocessing a load module deletes the END statement, use the linkage editor ENTRY control statement to specify the entry point of the new load module. For linkage editor requirements, refer to *MVS/ESA Linkage Editor and Loader*.

To change the name of an installed interpret table, do the following steps:

1. Change the name specified on the INTAB macroinstruction and follow the steps for changing an installed interpret table. When link-editing the interpret table, the module name and entry point must be made to match the new name of the interpret table.
2. Change the LOGTAB operand in all affected definition statements.
3. File the corrected NCP source program and the corrected set of definition statements in the appropriate VTAM library, replacing the old source programs. (The procedure is described in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* .) It is not necessary to do a partial NCP generation because the LOGTAB operand is used only by VTAM.
4. Use the MODIFY TABLE command to associate the new table with the corresponding logical unit, and load the table.

Full Syntax

The full syntax for the interpret table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

INTAB

```
►►──┬──────┬──INTAB──────────────────────────────────────────────────►◄
    └name──┘
```

LOGCHAR

```
►►──┬──────┬──LOGCHAR──APPLID=──┬─(APPLICID,application_name)──┬──────────────►
    └name──┘                    ├─(ROUTINE,routine_name)───────┤
                                └─(USERVAR,uservar_name)───────┘
```

```
►──┬─REMOVE=NO──────────┬──┬──────────────────────────────────────────────►◄
   │                    │  │
   └─REMOVE=──┬─NO──┬────┘  └─SEQNCE='characters'─┘
              └─YES─┘
```

ENDINTAB

```
►►──┬──────┬──ENDINTAB──────────────────────────────────────────────────────►◄
    └─name─┘
```

Operand Descriptions

The following section describes the operands you can code for the interpret table.

APPLID

```
►►──APPLID=──┬─(APPLICID,application_name)──┬──────────────────────────────►◄
             ├─(ROUTINE,routine_name)───────┤
             └─(USERVAR,uservar_name)───────┘
```

macroinstructions:	*LOGCHAR*
dependencies:	*none*

Specifies the name of an application program, a logon-interpret routine, or a USERVAR.

APPLID=(APPLICID,*application_name*)
Specifies the name of the application program. *application_name* can be any of the following:
- ACBNAME of an application program in this host
- *applname* of an application program in this host
- *applname* of an application program in another host
- USERVAR representing an application program

application_name can be a network-qualified name. A network-qualified name takes the form of *netid.application_name*. If *application_name* is network-qualified, then the network identifier is considered real and is not allowed to change. The resource name of the network-qualified name is considered generic and can undergo USERVAR translation.

Note: If ACBNAME and the network name on the APPL definition statement for the application program are different, then you cannot use a network-qualified ACBNAME.

APPLID=(ROUTINE,*routine_name*)
Specifies the routine name of the associated logon-interpret routine. All logon-interpret routines specified in an interpret table must be assembled and link-edited with that interpret table. Refer to *OS/390 SecureWay Communications Server: SNA Customization* for details on coding logon-interpret routines.

APPLID=(USERVAR,*uservar_name*)
The same as specifying APPLICID.

name

```
►►──┬──────┬──────────────────────────────────────────────────────►◄
   └name──┘
```

macroinstructions: *INTAB, LOGCHAR, ENDINTAB*
dependencies: *none*

Specifies optional names for the INTAB, LOGCHAR, and ENDINTAB macroinstructions. If specified, *name* must be unique and should be used as the operand for the assembler language END statement.

When the INTAB, LOGCHAR, and ENDINTAB macroinstructions are assembled, this name is used to identify the entry point to the interpret table CSECT. You can also use this name as a member name for this interpret table when you use the linkage editor to put the interpret table into the appropriate library. This assignment prevents different names from being used for the same interpret table (one for the entry point to the interpret table CSECT, and another as the name on the LOGTAB operand that other NCP and VTAM definition statements use to refer to this interpret table).

The interpret table is used by terminals that are defined by NCP and VTAM definition statements. The name assigned to the interpret table in the VTAM definition library must therefore be coded in the LOGTAB operand of those statements.

REMOVE

```
      ┌─REMOVE=NO──────┐
►►────┤                ├──────────────────────────────────────────►◄
      └─REMOVE=──┬─NO──┬─┘
                 └─YES─┘
```

macroinstructions: *LOGCHAR*
dependencies: *none*

Specifies whether VTAM is to remove the first nonblank set of characters from the user logon sequence before it passes the CINIT request unit (RU) to the application program.

REMOVE=YES
Specifies that VTAM is to remove the first nonblank set of characters from the user logon sequence data being processsed. The remaining data is left-justified and padded with blanks on the right. You can substitute Y for YES when coding this parameter.

REMOVE=NO
Specifies that VTAM is not to remove any data from the user logon sequence. You can substitute N for NO when coding this parameter.

For example, if the following information is sent and REMOVE=Y is specified, VTAM removes “IMS10” before it passes the information to the application program in the user data field of the CINIT RU.

```
IMS10 NAME PASSWORD   =====>    NAME PASSWORD
```

SEQNCE

```
►►──┬──────────────────────────────┬──────────────────────────────────────►◄
    └SEQNCE='characters'           ┘
```

macroinstructions:	*LOGCHAR*
dependencies:	*required for SNA devices*

Specifies the required part of a logical unit's logon message.

If '*characters*' is a logon message, optional information, which is not specified in the LOGCHAR macroinstruction, can be used by the logon-interpret routine (if the ROUTINE operand is specified), or by an application program's LOGON exit routine.

To specify an apostrophe (') or an ampersand (&) within the logon message, code a double apostrophe (''') or a double ampersand (&&) within the character string. If the terminal user enters the logon message in lowercase and the message is not translated to uppercase (for example, by USS translate table), the value for '*characters*' must be coded in lowercase.

Do not specify leading and trailing device-control characters within a character string that is to be interpreted, because the USS facility deletes these characters. Device control characters coded within a logon message will be deleted; therefore, a blank should not be coded for each occurrence of these characters. However, if a character within the logon message will be translated to a blank by the interpret table, code a blank to represent that character.

For non-SNA terminals, the length of the entire logon message (required information plus optional information) is limited to the number of characters that can be accommodated on one line of the terminal screen, up to 255 characters. For SNA terminals, the length is limited to 255 characters (VTAM deletes new-line [NL] characters before the INTRPRET macroinstruction is completed).

For non-SNA terminals, if you do not code SEQNCE in one LOGCHAR macroinstruction and a logon message does not match the character string of the SEQNCE operand in a preceding LOGCHAR macroinstruction in the interpret table, VTAM accepts this logon message and requests logon for this terminal to the application program specified in the LOGCHAR macroinstruction (the one in which SEQNCE is not coded). Therefore, do not place a LOGCHAR macroinstruction at the beginning of the interpret table (immediately following the INTAB macroinstruction) without coding the SEQNCE operand. Otherwise, the remaining logon messages in the interpret table are not compared with the logon message entered by the terminal user.

If you code LOGCHAR for SNA devices, you need to code SEQNCE. Otherwise, VTAM rejects an attempt to logon to the application specified in the APPLID operand when you press the enter key. VTAM issues a sense code stating the enter key RU is not valid because it is null length. Pressing the space key (an acceptable RU), then the enter key allows the logon.

Note: If you use two or more LOGCHAR macroinstructions, arrange them so that their SEQNCE fields are in reverse collating order.

Logon Mode Table

A logon mode table contains one or more sets of session parameters representing rules (session protocols) that describe how a session is to be conducted. You define each table by specifying the following macroinstructions:

- A MODETAB macroinstruction
- One or more MODEENT macroinstructions
- A MODEEND macroinstruction

VTAM has an IBM-supplied logon mode table named ISTINCLM that provides generally accepted session protocols for a basic list of IBM device types. See “Default Logon Mode Table (ISTINCLM)” on page 769 for the macroinstructions and operands that make up the default logon mode table.

For your information, the code released to you as the VTAM licensed program includes the source language for this table.

The source language is located in SYS1.ASAMPLIB.

You can modify or replace the IBM-supplied logon mode table, provided that the modified or replacement table has the same name as the IBM-supplied table and that the IBM-supplied table is deleted. However, you should create a supplemental table instead of deleting the IBM-supplied table, because you might need the IBM-supplied table for problem determination.

You can associate a logon mode table with a logical unit by specifying the table’s name in the MODETAB operand of the logical unit’s definition statement. If you do not designate a logon mode table for a logical unit, VTAM uses the IBM-supplied logon mode table, ISTINCLM. Furthermore, because you cannot code a resource-specific logon mode table for dynamic CDRSCs, VTAM uses the IBM-supplied logon mode table, ISTINCLM.

Regardless of the source of the logon or what session parameters are associated with the logon, the application program decides which session parameters are to be used for the session.

Table 76. Logon Mode Table Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	MODETAB					
name	MODEENT	APPNCOS	702			APPN only
		CKEY	702			
		COMPRES	703		SYSTEM	
		COMPROT	703		0	
		COS	704			
		DCODE	704			
		ENCR	704		0	data encryption
		ENCRTYP	705		0	data encryption
		FMPROF	706		0	
		LANG	706		0	
		LOGMODE	707			
		MAC	708			data encryption
		MACLNTH	709			data encryption

Table 76. Logon Mode Table Macroinstructions and Operands (continued)

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
		MACTYPE	709			data encryption
		PRIPROT	710		0	
		PSERVIC	710		0	
		PSNDPAC	710		0	
		RUSIZES	711			
		SECPROT	712		0	
		SRCVPAC	712		0	
		SSNDPAC	712		0	
		TSPROF	713		0	
		TYPE	713		1	
name	MODEEND					

Coding Notes

MODETAB Macroinstruction

The MODETAB macroinstruction indicates the beginning of a logon mode table.

MODEENT Macroinstruction

A MODEENT macroinstruction associates a logon mode name with a set of parameters representing session protocols.

The component description for the device represented by the logical unit should describe the features you want to use and the session protocols required to use those features. Refer to *OS/390 eNetwork Communications Server: SNA Programming* for information on setting session parameter fields. If a bit is identified as being reserved, set it to 0.

After you have determined what bits must be set, convert the eight bits in each byte of the session parameter field into the two equivalent hexadecimal digits. The value for each operand is the hexadecimal equivalent of the bits in the byte with which the operand is associated. For example, the operand PRIPROT is associated with byte 3 of the session parameter field. If you decide, for example, that the appropriate bit settings for the byte are 1111 0001, code PRIPROT=X'F1'.

The values for the FMPROF and TSPROF operands, can be specified as unframed decimal digits (for example, 4) or as framed hexadecimal digits (for example, X'04'). The other operands require hexadecimal values.

The MODEENT macroinstruction follows the MODETAB or another MODEENT macroinstruction.

MODEEND Macroinstruction

The MODEEND macroinstruction indicates the end of the logon mode table. Code it after the last MODEENT macroinstruction.

For Independent Logical Units

If the session partner of the logical unit for which this mode table applies is an independent logical unit, code only these operands:

- PSNDPAC
- RUSIZES
- SRCVPAC
- SSNDPAC

Installing a Logon Mode Table

Each logon mode table must be assembled and link-edited into a library associated with the VTAMLIB DD statement.

With the MODIFY TABLE command, the user can refresh or reload the logon mode table without deactivating the major nodes that should use the new table. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for information on using the MODIFY TABLE command.

Full Syntax

The full syntax for the logon mode table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

MODETAB

```
►►──MODETAB──────────────────────────────────────────────────────────►◄
```

MODEENT

```
►►──┬──────┬──MODEENT──┬──────────────────────────┬──┬──────────────────────────┬──►
    └name──┘           └APPNCOS=appn_cos_name─────┘  └CKEY=──┬─ALTERNATE─┬──────┘
                                                              └─PRIMARY───┘

►──┬─COMPRES=SYSTEM──────────┬──┬─COMPROT=0──────────────────┬──────────────────►
   │                         │  │                            │
   └COMPRES=──┬─SYSTEM──┬────┘  └COMPROT=common_lu_protocol──┘
              │         │
              ├─PROHIB──┤
              └─REQD────┘

                                                   ┌─ENCR=0──────┐
►──┬──────────────────────┬──┬──────────────────┬──┤             ├──────────────►
   └COS=subarea_cos_name──┘  └DCODE=device_code─┘  └ENCR=crypt───┘

   ┌─ENCRTYP=0──────┐  ┌─FMPROF=0──────────────────┐  ┌─LANG=0──────────────────┐
►──┤                ├──┤                           ├──┤                         ├──────────►
   └ENCRTYP=TDES24──┘  └FMPROF=function_mgt_prof───┘  └LANG=language_info──────┘
```

Logon Mode Table

```
►──┬──────────────────────────┬──┬──────────────┬──┬──────────────┬──►
   └LOGMODE=logon_mode_name───┘  │MAC=NONE      │  │MACLNTH=4     │
                                 │              │  │              │
                                 └MAC=─┬─NONE─┬─┘  └MACLNTH=─┬─4─┬─┘
                                       ├─COND─┤               ├─2─┤
                                       └─REQD─┘               ├─6─┤
                                                              └─8─┘

►──┬──────────────┬──┬──────────────────────┬──────────────────────►
   │MACTYPE=CRC   │  │PRIPROT=0             │
   │              │  │                      │
   └MACTYPE=─┬─CRC─┬─┘  └PRIPROT=plu_protocol┘
             └─DES─┘

►──┬──────────────────────────────┬──────────────────────────────►
   │PSERVIC=0                     │
   │                              │
   └PSERVIC=present_services_prof─┘

►──┬──────────────────────────────┬──┬──────────────────┬──────────►
   │PSNDPAC=0                     │  │RUSIZES=ru_size   │
   │                              │  │                  │
   └PSNDPAC=prim_send_pace_count──┘  └──────────────────┘

►──┬──────────────────┬──┬──────────────────────────────┬──────────►
   │SECPROT=0         │  │SRCVPAC=0                     │
   │                  │  │                              │
   └SECPROT=slu_protocol┘  └SRCVPAC=sec_receive_pace_count┘

►──┬──────────────────────────────┬──┬──────────────────────────┬──►
   │SSNDPAC=0                     │  │TSPROF=0                  │
   │                              │  │                          │
   └SSNDPAC=sec_send_pace_count───┘  └TSPROF=trans_services_prof┘

►──┬──────────┬──────────────────────────────────────────────────►◄
   │TYPE=1    │
   │          │
   └TYPE=bind─┘
```

MODEEND

```
►►──┬──────────┬──MODEEND──────────────────────────────────────►◄
    │          │
    └name──────┘
```

Operand Descriptions

The following section describes the operands you can code for the logon mode table.

APPNCOS

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─APPNCOS=appn_cos_name────┘
```

macroinstructions: *MODEENT*
dependencies: *APPN only*

Specifies the name of the APPN class of service to be used for the APPN portion of the session that uses this logon mode.

name can be 1–8 characters.

Class-of-service names are determined as follows:

- The subarea portion of the session uses the class-of-service name coded on the COS operand.
- The APPN portion of the session uses the class-of-service name coded on the APPNCOS operand.
- Using both the COS and APPNCOS operands enables you to use two different class-of-service names for the subarea and APPN portions of the same session.
- If the COS operand is coded but the APPNCOS operand is not coded, the COS name is used for both the subarea and APPN portions of the session. Therefore, if you want to use the same COS name to represent identical classes of service in both the subarea and APPN networks, coding the COS operand is sufficient. You do not need to code the APPNCOS operand.
- If the APPNCOS operand is coded but the COS operand is not coded, the default class-of-service name (8 blanks) is used for the subarea portion of the session and the APPNCOS class-of-service name is used for the APPN portion.
- If neither the COS operand nor the APPNCOS operand is coded, then the default class-of-service names (8 blanks for the subarea portion of the session and #CONNECT for the APPN portion) are used.

CKEY

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─CKEY=──┬─ALTERNATE─┬────┘
             └─PRIMARY───┘
```

macroinstructions: *MODEENT*

Specifies which cryptographic key name should be used for this logical unit. If you do not specify CKEY, any previous specification is unchanged.

ALTERNATE
Specifies that cryptographic session keys are generated using the alternate level of the cryptographic key name with the suffix ALT.

You can abbreviate ALTERNATE as ALT.

PRIMARY
Specifies that cryptographic session keys are generated using the primary level of the cryptographic key name (that is, the LU name on the LU definition statement or the value of the CKEYNAME).

You can abbreviate PRIMARY as PRIM.

COMPRES

```
►►─┬─COMPRES=SYSTEM──────────┬──────────────────────────────────────►◄
   │                         │
   └─COMPRES=─┬─SYSTEM──┬────┘
              │         │
              ├─PROHIB──┤
              └─REQD────┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies how the negotiation of data compression usage should be handled. This operand allows data-compression override specifications to be associated with a logon mode name. The SLU sends this information to the PLU to indicate what level of data compression it wants before the PLU sends the BIND.

COMPRES=PROHIB
Specifies that override data should explicitly prevent the use of data compression for sessions using this logon mode.

COMPRES=REQD
Specifies that override data should explicitly request data compression, regardless of link characteristics along the path, for sessions using this logon mode, if both session partners can compress data.

REQD causes compression to be specified only when you specify one of the following:

- The CMPVTAM start option is not set to 0
- One of the following is not set to 0 on the APPL definition statement:
  - The CMPAPPLI operand
  - the CMPAPPLO operand
  - Both the CMPAPPLI operand and the CMPAPPLO operand

If your application is the SLU and you want to use compression, code REQD.

COMPRES=SYSTEM
Specifies that no override is needed. For sessions using this logon mode, normal BIND negotiation logic determines whether data compression is used. The default system compression for VTAM application programs when VTAM acts as the SLU is no compression.

COMPROT

```
►►─┬─COMPROT=0──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─COMPROT=common_lu_protocol─┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Represents the common LU protocols (bytes 5 and 6 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0–FFFF can be specified.

COS

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─COS=subarea_cos_name─┘
```

macroinstructions: *MODEENT*
dependencies: *subarea only*

Specifies the name of the class of service to be used for a session that uses this logon mode. It should be a 1–8 character name. If COS is omitted, 8 blanks will be stored into the COS field in the mode entry. This blank name is the default COS.

Note: Because CP-CP session paths can include subarea VRs, it is strongly recommended that you specify an appropriate COS=*subarea_cos_name* in your logon mode tables (including the CPSVCMG and CPSVRMGR logon mode table entries in the IBM-supplied logon mode table, ISTINCLM). Otherwise, a blank COS name is used to determine the subarea VR and transmission priority that will be used for the VR portion of the CP-CP session path.

DCODE

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─DCODE=device_code────┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the LU device code, and indicates to application programs whether a teletypewriter exchange service (TWX) or a World Trade teletypewriter (WTTY) device is a keyboard and printer or a keyboard and display terminal.

Values for DCODE can be one of the following:
X'00' The terminal is a keyboard and printer.
X'80' The terminal is a keyboard and display.

If a value that is not valid is specified for DCODE, or if the parameter is not specified at all, the device is assumed to be a keyboard and printer.

ENCR

```
    ┌─ENCR=0──────┐
►►──┤             ├──────────────────────────────────────────────────►◄
    └─ENCR=crypt──┘
```

macroinstructions: *MODEENT*
dependencies: *VTAM data encryption facility only*

Specifies what type of cryptography the logical unit expects.

Note: ENCR can be specified only for the VTAM data encryption facility.

Any hexadecimal number in the range 0–F (or an equivalent decimal value) can be specified. One or two digit hexadecimal strings can also be used. A 4-bit binary string (for example, ENCR=B'0001') can also be specified.

The ENCR value is converted to a 4-bit string, which is treated as two 2-bit fields. The meanings of the bit settings are as follows:

xx.. Private cryptography field

00.. No private cryptography

01.. Private cryptography used

..xx VTAM cryptography field

..00 No session-level cryptography

..01 Selective cryptography; the primary logical unit can encipher messages and the secondary logical unit must support cryptography.

..10 Reserved

..11 Mandatory cryptography; all messages on this session will be enciphered and deciphered.

This value is stored in the first four bits of byte 26 in the session parameter field.

Note: The VTAM data encryption facility can use the following cryptography facilities:
- Integrated Cryptographic Service Facility (ICSF/MVS)
- IBM Programmed Cryptographic Facility program product (PCF)
- IBM Cryptographic Unit Support program product (CUSP)
- Translation Security System product (TSS)

ENCRTYP

```
►►─┬─ENCRTYP=0────────┬──────────────────────────────────────────►◄
   └─ENCRTYP=TDES24───┘
```

macroinstructions: *MODEENT*
dependencies: *VTAM data encryption facility only*

Specifies that the session using this log mode entry must utilize Triple DES encryption (if the session uses encryption at all). The only valid value that can be coded is TDES24 (Triple DES with 24—byte keys). If this operand is not coded, then DES encryption (the current level of cryptography) will be used if the session uses encryption at all. Triple DES is only viable with the Integrated Cryptographic Service Facility (ICSF/MVS).

FMPROF

```
►►─┬─FMPROF=0──────────────────┬──────────────────────────────────────────►◄
   └─FMPROF=function_mgt_prof──┘
```

macroinstructions:	*MODEENT*
dependencies:	*none*

Specifies the function management profile (byte 1 in the session parameter field) for this logon mode.

Any hexadecimal number in the range 0–FF (or its equivalent decimal value) can be specified. Values 2, 3, 4, 7, and 18 have defined meanings that are described in *OS/390 eNetwork Communications Server: SNA Programming*.

LANG

```
►►─┬─LANG=0──────────────────┬──────────────────────────────────────────►◄
   └─LANG=language_info──────┘
```

macroinstructions:	*MODEENT*
dependencies:	*none*
range:	0–255. or X'0–FF'

Defines the language information that will be passed to the VTAM application program. Refer to *OS/390 eNetwork Communications Server: SNA Programming* for details on how the application program receives this information.

Any decimal number in the range 0–255 can be specified. Equivalently, any hexadecimal number in the range 0–FF can be specified by using the standard assembler notation *number* .

TSO/VTAM uses the bits 0–7 when processing the value coded for the LANG operand:

- Bit 0

  TSO/VTAM uses this bit to indicate that devices with extended data stream capability should be queried for language information (double-byte character set capability). Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for additional information on how TSO/VTAM uses the entire value coded for LANG.

  The value of this bit always comes from the LANG operand of the MODEENT (it cannot be overridden using the LANG or LANGTAB operands on a USS command entered by a terminal user).
- Bit 1–7

  specifies the preferred language to be passed to the application program. The application program can use this information for language specific processing.

TSO/VTAM uses this language in several ways and it might affect the following:

- The language used for TSO/VTAM end-user messages

- The filtering done on single-byte character set (SBCS) data streams
- The language information returned on the GTTERM macroinstruction

For additional information on how TSO/VTAM uses this language, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

For a list of valid values to use for this field, refer to *OS/390 eNetwork Communications Server: SNA Programming* The value of this field can be overridden if the terminal operator enters a LOGON command with the LANG or LANGTAB operand. See "LOGON Command" on page 758 for more information. Refer to the *3274 Control Unit Description and Programmer's Guide* for the set of valid character codes for katakana. The 3274 Control Unit must be at microcode level 65 or higher to use the LANG operand.

Bit 0123 4567	Description
0...	The character set is identified from the code specified in bits 1–7. The device is assumed incapable of double-byte character set (DBCS) processing.
1...	A query command is sent to the device to determine the character set and DBCS capability. If the character set cannot be determined from the input received from the query, the character set is identified from the code specified in bits 1–7.
.000 0000 or .000 0001	The character set is U.S. English or any single-byte character set (SBCS) in which the valid character codes are compatible with the English codes.
.001 0001	The character set is katakana, or any SBCS in which the valid character codes are compatible with katakana.
.111 1111	VTAM performs no character set translation. Any SBCS in which the valid character codes are not compatible with English or katakana (for example, Thai) should specify this value.

Note: The only way to establish a kanji character-translation set (or alternate double-byte character set (DBCS)) is to query the device by specifying LANG=X'80' in the MODEENT macroinstruction. The correct character translation set for the device will be established from the device's response to the query command.

LOGMODE

```
►►──┬──────────────────────┬──────────────────────────────────────►◄
    └─LOGMODE=logon_mode_name─┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the logon mode name to be used as a key for the session parameters in this table entry.

When coding LOGMODE, consider the following information:

- If two or more blank (or omitted) LOGMODE entries exist in the same table:

  - The table assembles.
  - An assembler message warns you that blank entries were found.
  - The first blank entry in the table will be used (when a blank entry is going to be used). Subsequent blank entries in the table are ignored.
- If duplicate non-blank LOGMODE names exist in the same table, an assembler error informs you of a “previously defined symbol” and the table assembly fails.

  Correct the error (remove the duplicate logmode entry or use a different logmode name), and reassemble the table.
- Do not use the following symbols to specify LOGMODE names using the following symbols, which are used by the MODETAB, MODEENT and MODEEND macros to generate tables:

  ```
  ISTMnnnn  ISTSnnnn  ISTEnnnn    ISTXnnnn
  ```

  where `nnnn` is an integer from 1 to the number of LOGMODE tables (i.e., the number of MODETAB macros) contained in the LOGMODE table being assembled.

MAC

```
                 ┌─MAC=NONE──────────┐
►►───────────────┼──────────────────┼──────────────────────────────────────►◄
                 │          ┌─NONE─┐ │
                 └─MAC=─────┼──────┼─┘
                            ├─COND─┤
                            └─REQD─┘
```

macroinstructions: *MODEENT*

Specifies the message authentication requirements for this logical unit.

MAC=REQD
Specifies that all messages on this session must be authenticated. When MAC=REQD, a session cannot be activated unless the logical unit or host associated with the application program supports message authentication.

MAC=COND
Specifies that this application program supports message authentication, which is required if the session partner is MAC capable.

Note: If a cryptograpic product is not available at session start, a session without message authentication support and cryptographic enhancements is established.

MAC=NONE
Specifies that this logical unit does not engage in message authentication support and cryptographic enhancements supported sessions.

MACLNTH

```
┌─MACLNTH=4──────────┐
├──────────────────────────────────────────────────────────────►◄
└─MACLNTH=─┬─4─┬──┘
           ├─2─┤
           ├─6─┤
           └─8─┘
```

macroinstructions: *MODEENT*

Specifies the minimum length, in bytes, of the message authentication code that is to be generated.

MACLNTH is based on MACTYPE, and can be one of the following values:

MACTYPE	**MACLNTH**
CRC	2 or 4
DES	4, 6, or 8

MACTYPE

```
┌─MACTYPE=CRC────────┐
├──────────────────────────────────────────────────────────────►◄
└─MACTYPE=─┬─CRC─┬──┘
           └─DES─┘
```

macroinstructions: *MODEENT*

Specifies the method VTAM will use to create or verify a message authentication code.

MACTYPE=CRC
Specifies that VTAM use an internal cyclic redundancy check (CRC) algorithm to generate a message authentication code.

MACTYPE=DES
Specifies that VTAM use the data encryption standard (DES) services provided by the available Common Cryptographic Architecture (CCA) product to generate a message authentication code.

name

```
├──────────────────────────────────────────────────────────────►◄
└─name─┘
```

macroinstructions: *MODETAB, MODEENT, MODEEND*
dependencies: *none*

Logon Mode Table

Specifies optional names for the MODETAB, MODEENT, and MODEEND macroinstructions.

PRIPROT

```
►►─┬─PRIPROT=0──────────────┬──────────────────────────────────────────►◄
   └─PRIPROT=plu_protocol─┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the primary LU protocol (byte 3 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0–FF can be specified.

PSERVIC

```
►►─┬─PSERVIC=0──────────────────┬──────────────────────────────────────►◄
   └─PSERVIC=present_services_prof─┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the LU presentation services profile and LU presentation services usage field (bytes 13–24, respectively, in the session parameter field) for this logon mode. Specify a 24-digit hexadecimal number, using the bit settings described in *OS/390 eNetwork Communications Server: SNA Programming*.

PSNDPAC

```
►►─┬─PSNDPAC=0──────────────────┬──────────────────────────────────────►◄
   └─PSNDPAC=prim_send_pace_count─┘
```

macroinstructions: *MODEENT*
range: *0–63 or X'00'–X'3F'*

Specifies the primary send pacing count and whether one- or two-stage pacing is to be used. You can specify decimal or hexadecimal numbers. For decimal numbers, the range is 0–63. For hexadecimal numbers, the pacing count cannot be greater than X'3F'. However, any number in the range X'00'–X'FF' that does not use the reserved bit in position 1 can be specified. The low-order 6 bits of the hexadecimal number correspond to the primary pacing count. The high-order bit is used to specify whether one- or two-stage pacing is to be used. Note that the high-order bit can be set only by coding a hexadecimal value for this operand. Refer to *OS/390 eNetwork Communications Server: SNA Programming* for more information. Also see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

RUSIZES

```
►►──┬──────────────────┬──────────────────────────────────────────►◄
    └RUSIZES=ru_size───┘
```

macroinstructions: *MODEENT*
range: *see formula*

Specifies a portion of the transmission services usage field (bytes 9 and 10 in the session parameter field) for this logon mode. It specifies the maximum length in bytes of the RUs that the primary logical unit and secondary logical unit can send to each other, as calculated by the formula $m \times 2^n$.

Specify RUSIZES as four hexadecimal digits. The leftmost two digits apply to the secondary logical unit, and the rightmost two digits apply to the primary logical unit.

The format is the same for both sets of digits. The first digit is the mantissa (*m*), and it must be in the range X'8'–X'F'. The second digit is the exponent (*n*), and it must be in the range X'0'–X'F'.

For example, RUSIZES=X'96A8' specifies that the secondary logical unit can send a maximum length of 9×2^6 (or 576) bytes and that the primary logical unit can send a maximum of 10×2^8 (or 2560) bytes.

Table 77 shows the exponents, mantissas, and the values for RUSIZES.

Note: For non-APPC sessions, if both the mantissa and exponent are set to 0 or if RUSIZES is not specified, then the default size is used. The default is obtained as follows:

- If the BIND is an extended BIND, the default is X'FFFF'.
- If the BIND is not an extended BIND, the default is the value of the MAXLURU start option.

For APPC sessions, if the RU size value is zero or less than 256, the value is overridden and set to 256.

Table 77. Exponent and Mantissa Values for Each RU size

Exp (n)	Mantissa (m)							
	8	***9***	***A***	***B***	***C***	***D***	***E***	***F***
0	8	9	10	11	12	13	14	15
1	16	18	20	22	24	26	28	30
2	32	36	40	44	48	52	56	60
3	64	72	80	88	96	104	112	120
4	128	144	160	176	192	208	224	240
5	256	288	320	352	384	416	448	480
6	512	576	640	704	768	832	896	960
7	1024	1152	1280	1408	1536	1664	1792	1920
8	2048	2304	2560	2816	3072	3328	3584	3840
9	4096	4608	5120	5632	6144	6656	7168	7680

Table 77. Exponent and Mantissa Values for Each RU size (continued)

Exp (n)	Mantissa (m)							
A	8192	9216	10240	11264	12288	13312	14336	15360
B	16384	18432	20480	22528	24576	26624	28672	30720
C	32768	36864	40960	45056	49152	53248	57344	61440
D	65536	73728	81920	90112	98304	106496	114688	122880
E	131072	147456	163840	180224	196608	212992	229376	245760
F	262144	294912	327680	360448	393216	425984	458752	491520

SECPROT

```
►►─┬─SECPROT=0──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─SECPROT=slu_protocol───────┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the secondary LU protocol (byte 4 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0–FF can be specified.

SRCVPAC

```
►►─┬─SRCVPAC=0──────────────────────┬──────────────────────────────────►◄
   │                                │
   └─SRCVPAC=sec_receive_pace_count─┘
```

macroinstructions: *MODEENT*
range: *0–63 or X'00'–X'3F'*

Specifies the secondary receive pacing count and whether fixed or adaptive pacing is to be used. You can specify decimal or hexadecimal numbers. For decimal numbers, the range is 0–63. For hexadecimal numbers, the pacing count cannot be greater than X'3F'. However, any number in the range X'00'–X'FF' that does not use the reserved bit in position 1 can be specified. The low-order 6 bits of the hexadecimal number correspond to the secondary receive pacing count. The high-order bit is used to specify whether fixed or adaptive pacing is to be used. Note that the high-order bit can only be set by coding a hexadecimal value for this operand. Refer to *OS/390 eNetwork Communications Server: SNA Programming* for more information. Also see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

SSNDPAC

```
►►─┬─SSNDPAC=0──────────────────┬──────────────────────────────────────►◄
   │                            │
   └─SSNDPAC=sec_send_pace_count┘
```

macroinstructions: *MODEENT*

range: *0–63 or X'00'–X'3F'*

Specifies the secondary send pacing count and whether one- or two-stage pacing is to be used. You can specify decimal or hexadecimal numbers. For decimal numbers, the range is 0–63. For hexadecimal numbers, the pacing count cannot be greater than X'3F'. However, any number in the range X'00'–X'FF' that does not use the reserved bit in position 1 can be specified. The low-order 6 bits of the hexadecimal number correspond to the secondary pacing count. The high-order bit is used to specify whether one- or two-stage pacing is to be used. Note that the high-order bit can be set only by coding a hexadecimal value for this operand. Refer to *OS/390 eNetwork Communications Server: SNA Programming* for more information. Also see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

TSPROF

```
►►─┬─TSPROF=0──────────────────────┬──────────────────────────────────────►◄
   │                               │
   └─TSPROF=trans_services_prof────┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the transmission services profile (byte 2 in the session parameter field) for this logon mode. Any hexadecimal number in the range 0–FF (or its equivalent decimal value) can be specified. Values 2, 3, 4, and 7 have defined meanings that are described in *OS/390 eNetwork Communications Server: SNA Programming*.

TYPE

```
►►─┬─TYPE=1────┬──────────────────────────────────────────────────────────►◄
   │           │
   └─TYPE=bind─┘
```

macroinstructions: *MODEENT*
dependencies: *none*

Specifies the type of BIND command represented by this logon mode entry (bits 4–7 of byte 0 in the session parameter field). While only the values of 0 and 1 are defined, any hexadecimal value in the range 0–F can be specified. If 0 is specified, the logon mode entry represents session parameters that could be sent in a negotiable BIND. If 1 is specified, a non-negotiable BIND should be sent.

Note: LU 6.2 sessions require a negotiable bind.

Specifying TYPE=0 (indicating a negotiable BIND entry) does not cause VTAM to necessarily send a negotiable BIND to the secondary logical unit. The logon mode entry can indicate only that the secondary logical unit can support a negotiable BIND. The application program issuing the OPNDST macroinstruction must specify PROC=NEGBIND (in the node initialization block (NIB)) before VTAM will send a negotiable BIND command. For more information on negotiable BIND, see *OS/390 eNetwork Communications Server: SNA Programming*.

Message-Flooding Prevention Table

The message-flooding prevention facility identifies and suppresses duplicate messages that are issued in rapid succession. This reduces the possibility of duplicate messages flooding the operator console and concealing critical information. VTAM bases its suppression on the time interval since the message was last issued and on the similarity of variable text in the original and subsequent message.

For each candidate message, the message-flooding prevention table contains the criteria that must be met before VTAM suppresses duplicate messages and whether suppressed messages are sent to the hardcopy log. The suppression criteria include the amount of time between the original and subsequent messages, and an indication of which variable text fields are to be compared. If the message is reissued within the specified time interval and the specified variable text fields contain the same information, VTAM suppresses the message.

Table 78. Message-Flooding Prevention Table Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	FLDTAB					
name	FLDENT	MESSAGE	719	R		
		HARDCOPY	717		YES	
		LIST	718			see description
		TIME	719		30	
		VARTEXT	719		YES	
name	FLDEND					

Legend:

R Required operand.

Coding Notes

FLDTAB Macroinstruction

The FLDTAB macroinstruction begins the message-flooding prevention table.

FLDENT Macroinstruction

One or more FLDENT macroinstructions specify which messages are to be suppressed.

FLDEND Macroinstruction

The FLDEND macroinstruction ends the table.

Installing a Message-Flooding Prevention Table

Store the set of macroinstructions as a member of SYS1.VTAMLST. Use the member name as the name of the table.

No assembly or link-edit is required to install this table. The table is dynamically built when:

- VTAM is initialized, if the FLDTAB start option has been specified
- An operator issues a MODIFY TABLE command specifying the table name

The MODIFY TABLE command lets you replace or modify the message-flooding prevention table without affecting VTAM operations. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information about this command.

To change the IBM-supplied message-flooding prevention table:
1. Copy the IBM-supplied table (ISTMSFLD).
2. Rename the copy.
3. Modify the copy.

VTAM uses the internal version of the IBM-supplied message-flooding prevention table if either of the following is true:
- No table has been specified (FLDTAB start option not specified)
- The specified table (on the FLDTAB start option) has failed

You can display the name of the message-flooding prevention table by specifying ISTNOP, the name of the application program that represents the network operator, on the DISPLAY ID command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation*.

Considerations for Message-Flooding Prevention

The following are rules for suppressing messages through the message-flooding prevention table:

1. The message is recorded in the VTAM internal-trace table.
2. The message is constructed, but is not transmitted to the operator. The message might be routed to other areas. For example, if you coded PPOLOG=YES in your start options, the message will be routed to the PPO (primary program operator) log.
3. If the first line of a message group is suppressed, all messages in the group are also suppressed.
4. A message within a message group cannot be suppressed by specifying that message on the FLDENT macroinstruction.
5. USSMSG00 – USSMSG14 are not suppressed.

The message-flooding prevention process can cause some undesirable circumstances to occur. For example:

1. A command might be entered and result in a message to the operator. If the message is a member of the flooding-prevention table and the command is reentered within the designated time-span, the message might not appear the second time as expected.
2. If the header of a message group is suppressed, all messages in the group are also suppressed. The information in these messages might differ from the last issuance, but they are not reissued.

Example of Message Suppression

The following example demonstrates when a message will be suppressed to prevent flooding of the operator console. This suppression occurs without regard to

the suppression level assigned to the message in the USS table and without regard to the message suppression level assigned to VTAM using the MODIFY SUPP command or the SUPP start option.

Message IST367I is a member of the message-flooding prevention table:

```
IST367I NO STORAGE TO DEFINE NODE nodename CONFIG configname
```

This message is issued when VTAM does not have enough storage to expand the host-node table. Therefore, a new host element address could not be assigned for the defined resource *nodename* in the configuration *configname*. The specifications for the message in the table are:

1. Time-span: 30 seconds
2. Variable fields: *configname*

Without message flooding prevention, this message is displayed every time it is issued without regard to time-span or *configname*.

```
Time  1:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG1
Time  2:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG2
Time  3:  IST367I  NO STORAGE TO DEFINE NODE NODE2 CONFIG CONFIG1
Time 35:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG1
```

With message flooding prevention, this message at Time 3 would not be displayed for the following reasons:

- *configname* is the same as *configname* at Time 1.
- The message at Time 3 was issued within 30 seconds of the message at Time 1.

The message at Time 2 would be displayed since *configname* is not the same.

The message at Time 35 would be displayed since it was issued after the 30 second time-span.

```
Time  1:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG1
Time  2:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG2
Time 35:  IST367I  NO STORAGE TO DEFINE NODE NODE1 CONFIG CONFIG1
```

Full Syntax

The full syntax for the message-flooding prevention table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

FLDTAB

```
►►──┬──────┬──FLDTAB──────────────────────────────────────────────────►◄
    └name──┘
```

FLDENT

```
►►──name──FLDENT──MESSAGE=message_id──┬──────────────────────┬──┤ HARDCOPY ├──────►
                                      └──LIST=(──msg_id──)───┘
```

```
►─┬─TIME=30──────────────────┬─┬─VARTEXT=YES──────────┬──────────────────────►◄
  └─TIME=number_of_seconds─┘ └─VARTEXT=(─┬─YES─┬─)─┘
                                         └─NO──┘
```

HARDCOPY

```
├─┬─HARDCOPY=YES──────┬──────────────────────────────────────────────────┤
  └─HARDCOPY=─┬─YES─┬─┘
              └─NO──┘
```

FLDEND

```
►►─┬──────┬─FLDEND──────────────────────────────────────────────────────►◄
   └─name─┘
```

Operand Descriptions

The following section describes the operands you can code for the message-flooding prevention table.

HARDCOPY

```
►►─┤ HARDCOPY ├──────────────────────────────────────────────────────────►◄
```

HARDCOPY

```
├─┬─HARDCOPY=YES──────┬──────────────────────────────────────────────────┤
  └─HARDCOPY=─┬─YES─┬─┘
              └─NO──┘
```

macroinstruction: *FLDENT*

Specifies whether a suppressed message is sent to the system hardcopy log.

HARDCOPY=YES
Indicates that the message is suppressed from the operator console, but is sent to the system hardcopy log.

HARDCOPY=NO
Indicates that the message is suppressed from the operator console and the system hardcopy log.

LIST

```
►►──┬──────────────────────────────┬──────────────────────────────────────────►◄
    └─LIST=(─msg_id──)─────────────┘
```

macroinstruction: *FLDENT*
dependencies: *see description*

Specifies from one to five messages that are linked with the first message in a group to determine whether the message group should be suppressed. The messages you specify on LIST must be present as FLDENT entries in the message-flooding prevention table.

With the LIST operand, you can prevent messages that are not duplicates from being suppressed. For instance, if you specify message IST663I on the MESSAGE operand, and do not code the LIST operand, then all message groups beginning with IST663I are suppressed, even if the groups are not duplicates. An example of this problem follows:

```
IST663I  CDINIT     REQUEST FROM IBMGW    FAILED  , SENSE=087D0001
IST664I  REAL  OLU=USIBMEL.ELNA9171    REAL  DLU=IBMIN.IBMN2CC9
(intervening messages removed)
IST663I  CDINIT     REQUEST FROM IBMGW    FAILED  , SENSE=087D0001
IST664I  REAL  OLU=USIBMP0.TF2LAADV    REAL  DLU=IBMIN.IBMJ1EP8
(intervening messages removed)
IST663I  CDINIT     REQUEST FROM IBMGW    FAILED  , SENSE=087D0001
IST664I  REAL  OLU=APPN.IKPASS0A       REAL  DLU=APPN.IKPASSHQ
(intervening messages removed)
```

In the above example, the three IST663I messages are treated as duplicates. The first IST663I multi-line message is kept and the others suppressed. However, the messages are not duplicates because they have different LU partner names in the IST664I messages.

If the second and third IST663I messages were suppressed, it would take you longer for problem determination, due to having to search trace tapes to find the failures that would have been displayed through the message.

But, if you were to specify IST664I on the LIST operand (LIST=0664) of the FLDENT entry for IST663I, VTAM would evaluate the IST664I messages for duplication, and not suppress them because they are not duplicates.

Notes:

1. The values of the LIST operand are the message IDs of the other FLDENT entries (for instance, LIST=0664). All LIST references must be to other messages with corresponding FLDENT entries in the message-flooding prevention table.
2. A LIST entry which is the same as its FLDENT entry is not permitted.
3. If the FLDENT entry indicates a LIST, but the message is being issued as a single-line message, the FLDENT entry is ignored for the one message.
4. If the FLDENT entry indicates a LIST with multiple messages and all of the messages are not present in the message group being issued by VTAM, the FLDENT entry is ignored for the current group.
5. If the FLDENT entry indicates a LIST with multiple messages, the messages do not have to be present within the multiple line write-to-operator group in the order indicated in the FLDENT entry.

6. You cannot specify a particular message in more than one LIST operand.
7. The HARDCOPY and TIME operands are ignored if either operand is coded for a message that is not a single-line message or the first message within a group. For the ″later messages″, the HARDCOPY and TIME values for the first message apply to the entire group.

MESSAGE

```
►►──MESSAGE=message_id──────────────────────────────────────────────────►◄
```

macroinstruction: *FLDENT*

Specifies a 1-digit to 4-digit identifier that corresponds to the message number to be suppressed. You can pad an identifier that has fewer than four digits with leading zeroes. For example, MESSAGE=118 and MESSAGE=0118 are viewed as the same.

Note: Messages do not have to be in numerical order in the table. However, for better performance, list the most common messages first.

name

```
►►──┬──────┬──────────────────────►◄   ►►──name──────────────────────►◄
    └─name─┘
```

macroinstructions: FLDTAB, FLDEND *macroinstructions:* FLDENT

Specifies the name of the macroinstruction being defined.

For FLDTAB and FLDEND: *name* is optional and has no function. If used, it is recommended that *name* match the SYS1.VTAMLST member name containing these macroinstructions.

For FLDENT: Specifies the required and unique 1–8 character name of the table entry.

TIME

```
►►──┬─TIME=30──────────────────┬──────────────────────────────────────►◄
    │                          │
    └─TIME=number_of_seconds───┘
```

macroinstruction: *FLDENT*
range: *1–32767*

Specifies the amount of time, in seconds, that VTAM continues to suppress duplicate messages. The time interval begins when the first message is issued.

VARTEXT

Message-Flooding Prevention Table

```
►►─┬─VARTEXT=YES──────────┬──────────────────────────────────────►◄
   │                      │
   └─VARTEXT=(─┬─YES─┬─)──┘
               └─NO──┘
```

macroinstruction: *FLDENT*

Specifies which variable text fields in the specified message are used as the criteria for suppression.

Code a VARTEXT setting or allow the default for each variable text field in the candidate message. For example, if you code

```
VARTEXT=(NO,,NO)
```

the first and third variable text fields in the message are not used as criteria for suppression. However, the second field is used because the default value YES is used.

The number of variable text fields can be determined by referring to *OS/390 SecureWay Communications Server: SNA Messages*.

VARTEXT is not applicable for messages that have no variable text. If the candidate message does not have variable text, and you code VARTEXT, VTAM issues an error message.

VARTEXT=YES
Indicates that VTAM compares the field's value in the duplicate message to the field's value in the original message when determining whether to suppress the duplicate message. If the fields contain different values, the message is not suppressed.

VARTEXT=NO
Indicates that VTAM does not compare the field values when determining whether to suppress the duplicate message.

Model Name Table

The model name table contains model names that can be passed to VTAM application programs in their LOGON exits. VTAM application programs use the model names to create dynamic definitions for their session-partner resources.

A model definition is usually the starting point for a dynamic resource definition. The model name passed by VTAM enables the application program to select the proper model definition to use.

Operands on an SLU's resource definition associate that SLU with the proper model name data. The MDLTAB operand specifies the name of the model name table to be used, and the MDLENT operand specifies the proper entry within the table. These operands are described for each LU-type resource in "Major Nodes" on page 19.

Table 79. Model Name Table Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	MDLTAB						
name	MDLENT	MODEL	722				
name	MDLPLU	PLU	723	R			
		MODEL	722				

Legend:

R Required operand.

Coding Notes

MDLTAB Macroinstruction

The MDLTAB macroinstruction indicates the beginning of the model name table.

MDLENT Macroinstruction

The MDLENT macroinstruction indicates the start of a model name entry and optionally sets up the default model name. The end of the entry is indicated either by another MDLENT macroinstruction or by an end of file.

MDLPLU Macroinstruction

The MDLPLU macroinstruction defines model name data for a specific PLU within the table entry started by the previous MDLENT macroinstruction.

Installing a Model Name Table

Store the model name table macroinstructions in SYS1.VTAMLST, and use the member name as the name of the table.

No assembly or link-edit is required to install the table. The table will be dynamically built upon the first activation of any resource that has a defined association to the table. When the last resource using it is deactivated, the table is deleted and its storage is freed.

IBM does not supply a default model name table.

Full Syntax

The full syntax for the model name table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

MDLTAB

```
►►──┬──────┬──MDLTAB──────────────────────────────────────────────────►◄
    └name──┘
```

MDLENT

```
►►──name──MDLENT──┬──────────────────────┬──────────────────────────────────►◄
                  └─MODEL=model_name─────┘
```

MDLPLU

```
►►──┬──────┬──MDLPLU──PLU=plu_name──┬──────────────────────┬──────────────►◄
    └─name─┘                        └─MODEL=model_name─────┘
```

Operand Descriptions

The following section describes the operands you can code for the model name table.

MODEL

```
►►──┬──────────────────────┬──────────────────────────────────────────────►◄
    └─MODEL=model_name─────┘
```

macroinstructions: *MDLENT, MDLPLU*

Specifies the 1–8 character model name expected by the subsystem for the terminal.

The model name is only passed when VTAM initiates the session with the application program, for example, when the SLU has the LOGAPPL = parameter coded in its definition. When the SLU initiates the session with the application program by means of the LOGON command or a USS command, the model name to be passed to the application program must be explicitly specified on the MODEL = parameter of the LOGON command or USS command.

If you code MODEL on the MDLENT macroinstruction, the *model_name* specified is a default to be used with any PLU that is not specified in a following MDLPLU macroinstruction.

If you omit MODEL on the MDLENT macroinstruction, no default data is created. Consequently, any PLU not specified in one of the MDLPLU macroinstructions within this entry will not receive any model name data unless the MODEL operand is overridden during logon.

If you omit MODEL on the MDLPLU macroinstruction, no *model_name* is sent to the application or to the subsystem during session initiation.

name

```
►►──┬──────┬──────────────────────►◄   ►►──name──────────────────────────►◄
    └─name─┘
```

macroinstructions: MDLTAB, MDLPLU *macroinstructions:* MDLENT

Specifies the 1–8 character name for the macroinstructions.

PLU

```
►►──PLU=plu_name──────────────────────────────────────────────────►◄
```

macroinstructions: *MDLPLU*

Specifies the 1–8 character name of the PLU as it is known in the SLU's network. cms

Subarea Mapping Table

The subarea mapping (SAMAP) table is used to supply additional NCP connectivity information to APPN Topology and Routing Services, which can then use that information in an effort to optimize routes through the CNN. Refer to the OS/390 SecureWay Communications Server: SNA Operation and OS/390 SecureWay Communications Server: SNA Network Implementation Guide for more information on Subarea Mapping Tables.

Figure 5 illustrates definition of a SAMAP mapping table.

```
SAMAP1   VBUILD    TYPE=SAMAP
SA10     MAPSTO    SUBAREA=SA20
SA11     MAPSTO    NETID=NETB
```

Figure 5. Subarea Mapping Table Example

Table 80. Definition Statements for SAMAP Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=SAMAP	725	R			
SAxx	MAPSTO	SUBAREA=SAxx NETID=netid	724	R			

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code a VBUILD definition statement as the first statement in a subarea mapping table.

MAPSTO Definition Statement

Code a MAPSTO definition statement to define a subarea mapping.

Full Syntax

The full syntax for the subarea mapping table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────────┬──VBUILD──TYPE=SAMAP──────────────────────────────────►◄
    └─name─────┘
```

MAPSTO

```
►►──SA──xx──MAPSTO──=──┬──────────────────────┬──────────────────────────►◄
                       ├─SUBAREA──=──SAxx─────┤
                       └─NETID──=──netid──────┘
```

Operand Descriptions

The following section describes the operands you can code for the subarea mapping table.

name

```
►►──┬──────────┬──────────────────────────────────────────────────────►◄
    └─name─────┘
```

statements:	*VBUILD*
dependencies:	*none*

Assigns a name to the subarea mapping table.

NETID

```
►►──NETID──=──netid──────────────────────────────────────────────────►◄
```

statements:	*MAPSTO*
dependencies:	*none*

Assigns a name to the NETID.

SAxx

```
►►──SA──xx───────────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a decimal CNN subarea number. The SA prefix is required.

SUBAREA

```
►►──SUBAREA──=──SAxx──────────────────────────────────────────────────────►◄
```

statements: *MAPSTO*
dependencies: *none*

Specifies a subarea number.

TYPE

```
►►──TYPE=SAMAP──────────────────────────────────────────────────────────►◄
```

statements: *VBUILD*
dependencies: *none*

Means that this VBUILD definition statement defines a subarea mapping table.

Session Awareness (SAW) Data Filter

VTAM provides a filter to reduce the amount of session awareness (SAW) data that is passed to communication network management (CNM) application programs, such as the NetView program. Using the SAW data filter, only data for sessions that match predefined PLU-SLU name combinations are sent over the CNM interface to the CNM application program. The PLU-SLU combination definitions can contain pattern-matching characters so that one definition can apply to a variety of sessions.

Note: If the VTAM SAW data filter definition contains SSCP-PU SAW data filter definitions, the CNM application (for example, the NetView program) does not receive any SSCP-PU SAW data from VTAM and cannot collect response time monitor (RTM) data for active physical units.

The KCLASS and MAPSESS macroinstructions define the session monitor filter. If you use the NetView program, you can copy your existing session monitor filter definitions and make some changes to create equivalent VTAM filters. You must place the KEEPMEM START macroinstruction at the beginning of the filter definitions and add KEEPMEM STOP and END macroinstructions at the end of the existing definitions. For continuation lines, use the conventions described in “Rules for Coding Comments” on page 12. VTAM ignores any NetView operands that do not apply to the VTAM SAW data filter.

VTAM includes a default filter, ISTMGC10 in VTAMLIB, that allows data for all sessions to be passed across the CNM interface. ISTMGC10 needs to be assembled. See “Restrictions on Use of Assembler Features” on page 15 for information on assembling modules.

SAW Data Filter

You can modify ISTMGC10 or replace it with one of your own using the MODIFY TABLE command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation* for more information on the MODIFY TABLE command.

You can display the name of the SAW data filter table by specifying ISTNOP, the name of the application program that represents the network operator, on the DISPLAY ID command. Refer to the *OS/390 SecureWay Communications Server: SNA Operation*.

Table 81. Session Awareness Data Filter Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	KEEPMEM	START-STOP	729	R		
name	KCLASS	SAW	728	R	YES	
name	MAPSESS	KCLASS	727	R		
		PRINET	728			
		PRI	728			
		SEC	729			
		SECNET	729			
name	END					

Legend:

R Required operand.

Coding Notes

KEEPMEM Macroinstruction

The KEEPMEM macroinstruction defines the beginning or end of a set of SAW data filter definition macroinstructions.

KCLASS Macroinstruction

The KCLASS macroinstruction defines whether VTAM passes SAW data over the CNM interface for a particular session. The MAPSESS macroinstruction refers to the names on the KCLASS macroinstructions to instruct VTAM on how to treat SAW data for sessions that match the MAPSESS PLU-SLU name specifications.

MAPSESS Macroinstruction

The MAPSESS macroinstruction specifies a PLU-SLU name combination and specifies which KCLASS definition VTAM should use to determine if SAW data should be passed over the CNM interface.

Notes:

1. The maximum number of MAPSESS macroinstructions you can code is 2000.
2. If the PRI operand has the host CDRM name coded, an additional MAPSESS macroinstruction with PRI=VTAM must be specified. PRI=VTAM is used to obtain saw data before the activation of the host CDRM major node.

END Macroinstruction

The END macroinstruction indicates the end of a SAW data filter.

Full Syntax

The full syntax for the session awareness data filter follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

KEEPMEM

```
►►──name──KEEPMEM──┬─START─┬──────────────────────────────────────────►◄
                   └─STOP──┘
```

KCLASS

```
                          ┌─YES─┐
►►──name──KCLASS──SAW=────┤     ├──────────────────────────────────────►◄
                          └─NO──┘
```

MAPSESS

```
►►──┬──────┬──MAPSESS──KCLASS=kclass_name──┬──────────────────────────┬──►
    └─name─┘                               └─PRINET=plu_network_id────┘

►──┬──────────────────────┬──┬──────────────────────┬──┬──────────────────────┬──►◄
   └─PRI=pluname_pattern──┘  └─SEC=sluname_pattern──┘  └─SECNET=slu_network_id─┘
```

END

```
►►──┬──────┬──END──────────────────────────────────────────────────────►◄
    └─name─┘
```

Operand Descriptions

The following section describes the operands you can code for the saw data filter.

KCLASS

```
►►──KCLASS=kclass_name──────────────────────────────────────────────────►◄
```

macroinstructions:	*MAPSESS*
dependencies:	*none*

Specifies the name of the preceding KCLASS macroinstruction. This operand indicates whether SAW data is to be transmitted over the CNM interface for a particular session.

name

```
►►──┬──────┬──────────────────────────────►◄  ►►──name──────────────────────────────►◄
    └name──┘
```

macroinstructions: MAPSESS, END *macroinstructions:* KEEPMEM, KCLASS

Specifies 1–8 character names for the macroinstructions.

PRINET

```
►►──┬──────────────────────┬──────────────────────────────────────────────────►◄
    └PRINET=plu_network_id─┘
```

macroinstructions: *MAPSESS*
dependencies: *none*

Specifies a network ID for the PLU.

PRI

```
►►──┬──────────────────┬──────────────────────────────────────────────────────►◄
    └PRI=pluname_pattern─┘
```

macroinstructions: *MAPSESS*
dependencies: *none*

Specifies the pattern used to match the PLU name. It can contain 1–8 characters from the set (A–Z, 0–9, @, $, ?, *). A question mark (?) matches any single letter or digit. An asterisk (*) matches more than one letter or digit, and can be used only as the last character in the string. The pattern can be used to compare names only up to 8 characters in length.

Following are some examples:
- T3?7* matches T3A7, T3B7A, or T3C7ABCD. It does not match T3AA7A.
- T3*7? is not a valid pattern.
- T3??7 matches T3AB7 or T3CD7. It does not match T3A7.

SAW

```
►►──SAW=──┬─YES─┬──────────────────────────────────────────────────────────────►◄
          └─NO──┘
```

macroinstructions: *KCLASS*
dependencies: *none*

Specifies whether to pass SAW data over the CNM interface.

SEC

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─SEC=sluname_pattern──────┘
```

macroinstructions: *MAPSESS*
dependencies: *none*

Specifies the pattern used to match the SLU name. See the description of the PRI operand for details on the pattern-matching rules.

Note: SEC and PRI are optional. If one is not specified, the default value of “*” is taken.

SECNET

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └─SECNET=slu_network_id────┘
```

macroinstructions: *MAPSESS*
dependencies: *none*

Specifies a network ID for the SLU.

START-STOP

```
►►──┬─START─┬──────────────────────────────────────────────────────────────►◄
    └─STOP──┘
```

macroinstructions: *KEEPMEM*
dependencies: *none*

Specifies whether this macroinstruction starts or stops a set of session awareness data filter definition statements.

SAW Sense Filter

The SAW sense filter contains a list of session initialization failure sense codes that are not to be reported to a communication network management (CNM) application program collecting SAW data.

The SNSCODE definition statement defines the sense codes that are excluded from the SAW data being reported to the CNM application program. For more information about how to use the SAW sense filter, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* .

SAW Sense Filter

Table 82. SAW Sense Filter Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=SNSFILTR	731	R			
name	SNSCODE	SENSE	731				

Legend:

R Required operand.

Coding Notes

VBUILD Definition Statement

Code one VBUILD TYPE=SNSFILTR definition statement to indicate that this set of definition statements is for a SAW sense filter.

SNSCODE Definition Statement

Code one SNSCODE definition statement for each session initiation failure sense code to be filtered from SAW data sent to a communication network management (CNM) application program.

Full Syntax

The full syntax for the SAW sense filter follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

```
►►──┬──────┬──VBUILD──TYPE=SNSFILTR──────────────────────────────────►◄
    └name──┘
```

SNSCODE

```
►►──┬──────┬──SNSCODE──SENSE=sense_code──────────────────────────────►◄
    └name──┘
```

Operand Descriptions

The following section describes the operands you can code for the SAW data filter.

name

```
►►──┬──────┬──────────────────────────────────────────────────────────►◄
    └name──┘
```

Specifies the optional name of definition statement.

SENSE

```
►►──┬──────┬──────────────────────────────────────────────────────►◄
    └name──┘
```

statement: *SNSCODE*
dependencies: *none*

Specifies the session initiation failure sense code that is to be filtered out of the SAW data sent to CNM application program receiving SAW data. Specify either the two-byte or four-byte session initiation failure sense code (in hexadecimal). If one or three bytes of data is specified, the data is ignored. Any sense code specified that is not related to session initiation failures is also ignored.

Note: If only two bytes are specified, the whole category specified by the two bytes will be filtered.

TYPE

```
►►──TYPE=SNSFILTR──────────────────────────────────────────────────►◄
```

statement: *VBUILD*
dependencies: *none*

Indicates that this set of definition statement defines a SAW sense filter table.

Unformatted System Services (USS) Tables

VTAM uses unformatted system services (USS) tables to:

- Issue messages to the VTAM operator console or a program operator application
- Process VTAM operator commands
- Issue messages to a terminal operator user
- Process commands received from a terminal operator user

Other messages, such as TSO/VTAM messages, are provided through other facilities. For an explanation of these messages, refer to *OS/390 SecureWay Communications Server: SNA Messages*.

VTAM uses two types of USS tables to define messages and commands:

Session-Level Table

Contains:

- Definitions for terminal user commands (such as LOGON) that can be received from a terminal
- Messages that VTAM sends to a terminal
- A translation table that is used for character-coded input from the terminal.

See “Session-Level USS Table (ISTINCDT)” on page 784 for an example of a session-level USS table.

Operation-Level Table
Contains USS commands (such as DISPLAY ROUTE) that can be received from the VTAM operator or a program operator application and messages issued in response to those commands. See “Operation-Level USS Table (ISTINCNO)” on page 785 for an example of an operation-level USS table.

IBM supplies default USS tables with the VTAM product. The source language for these tables is contained in SYS1.ASAMPLIB. Table 83 describes the default tables.

Table 83. Description of the IBM-Supplied USS Tables

Table Name	Description
ISTINCDT	Default session-level USS table used by terminal operator users. Contains the following: • The following terminal operator commands: – LOGON – LOGOFF – IBMTEST • USSMSGxx (where xx equals 1–9 and 11–14) that are sent to a terminal operator. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for detailed descriptions of USS messages. • Translation table that is used for character-coded input from the terminal. The translation table, named STDTRANS, converts lowercase characters to uppercase and converts horizontal tabs to spaces.
ISTINCNO	Default operation-level USS table used by the VTAM operator (commands and messages) and program operator applications (commands). Contains the following: • Operands and defaults for the following VTAM operator commands: – All DISPLAY commands – All MODIFY commands – All VARY commands For more details on these commands, refer to *OS/390 SecureWay Communications Server: SNA Operation*. • VTAM operator messages listed in the *OS/390 SecureWay Communications Server: SNA Messages*.
ISTCFCMM	Default operation-level USS table for a program operator application containing operation-level VTAM messages.

Customizing USS Tables

You can customize VTAM messages, commands, and the translation table by coding your own USS table. For example, you can change messages to provide non-English text, change characteristics of a message, or change the syntax or default values for a command.

Any changes to VTAM messages or commands should be made with supplementary user-defined USS tables. The IBM-supplied USS tables should not be changed or removed since they are used to define all commands and messages that are not defined in a user-written table. The name of the user-defined USS table should be different from any of the default tables.

If you plan to make extensive changes in your user-defined USS table, you might find that the easiest way to generate the table is to copy the IBM-supplied USS tables and make changes to the new copies.

Note: Because some program operator applications depend on VTAM messages, changes to an operator message could disrupt the functioning of a program operator application.

A USS table can be associated with a particular resource by the following:

Terminal Operator
- LANGTAB operand on a USS command
- USSTAB operand on the logical unit's definition statement

VTAM Operator
- USSTAB start option
- MODIFY TABLE command

  For more information on the MODIFY TABLE command, refer to *OS/390 SecureWay Communications Server: SNA Operation*.

Program Operator
- USSTAB operand on the application definition statement
- SSCPFM operand on the application definition statement

VTAM Messages

To customize VTAM messages, you only need to code the message or the part of the message you wish to redefine. VTAM will use the user-defined supplementary USS message table in conjunction with the IBM-supplied default USS tables to locate the complete message definition. Several USS tables might be searched before locating a complete message definition. The tables are searched in a specific order (see "Order of USS Table Use").

For more information on coding VTAM messages in a USS table, see "Customizing Messages" on page 753.

VTAM Commands

Although a VTAM message need only be partially defined in a supplementary user-defined table, a command must be completely defined in a supplementary user-defined USS table. VTAM uses either the user-defined table or the IBM-supplied default USS tables to locate the complete command definition.

For more information on coding VTAM commands in a USS table, see "Defining VTAM Operator Commands" on page 756.

Order of USS Table Use

When VTAM requires a message, command, or input translation table that is defined in the USS tables, it searches the USS tables in a specific order. The search order depends on the destination or source (VTAM operator, program operator, or terminal operator) and the target search item (a command, message, or translation table).

Tables 84 through 87 show the search order used by VTAM for each target search item (messages, commands, and translation tables). Within each table the search order is shown for the first table searched to the last (top to bottom) for each destination (terminal operator, VTAM operator, and program operator).

For VTAM Messages

VTAM uses the search order shown in Table 84.

Table 84. VTAM Messages Search Order for VTAM Messages

Terminal Operator	VTAM Operator
MVS Message Service[1]	USSTAB START option
LANGTAB USS table	ISTINCNO
LU's definition	
ISTINCDT	

1. You can use this service to define USS messages for terminal operators. Refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* for information on how the MVS Message Service can be used to define USS messages for terminal operators.

For VTAM Messages for a Program Operator

VTAM uses the search order shown in Table 85.

Table 85. VTAM Messages Search Order for a Program Operator

Program Operator SSCPFM=USSNOP	Program Operator SSCPFM=USSPOI
APPL definition statement	APPL definition statement
ISTINCNO	ISTCFCMM
	ISTINCNO

If no USS message is found, VTAM will issue USS message 14 to indicate that the message was not found. (USS message 14 is defined the same as IST458I in ISTINCNO.)

Note: Do not modify or replace the IBM-supplied default tables.

If no VTAM message is found, VTAM issues message IST998E to indicate that a VTAM message is not defined.

A VTAM message might be rerouted from the program operator application to the system console. A rerouted message will appear in the format as defined in the USS table designated for the VTAM operator based on the search order.

If a VTAM message is not completely defined in the first USS table searched, the next USS table in the hierarchy will be used. Because the IBM-supplied USS table (ISTINCNO) contains complete messages, you should define supplemental operation-level USS tables to specify only those parts of a message that you want to change. If VTAM cannot locate all the required parts of a message, it issues message IST998E to indicate that the message does not exist.

For VTAM Commands

VTAM uses the search order shown in Table 86.

Table 86. VTAM Commands Search Order for VTAM Commands

Terminal Operator	VTAM Operator	Program Operator
LANGTAB USS table	USSTAB START option	APPL definition
LU's definition	ISTINCNO	ISTINCNO

Table 86. VTAM Commands Search Order for VTAM Commands (continued)

Terminal Operator	VTAM Operator	Program Operator
ISTINCDT		

If the USS command entered by the operator is not found, VTAM uses the command as entered.

For Input Translation Tables

VTAM uses the search order shown in Table 87.

Table 87. Input Translation Tables Search Order

Terminal Operator	VTAM Operator	Program Operator
LU's definition	USSTAB START option	APPL definition
ISTINCDT		

If no input translation table is found, VTAM performs no translation.

Installing a User-Defined USS Table

Each USS table must be assembled and link-edited as a non-executable module into a library associated with the SYS1.VTAMLIB DD statement.

Coding a User-Defined USS Table

To code a supplementary USS table, create a module using the following macroinstructions:

- USSTAB
- USSCMD
- USSPARM
- USSMSG
- USSEND

If a user-defined table is coded as part of another module, code an assembler EXTRN definition statement for the table name in that module so the table will be known externally and can be accessed by other modules.

If a character translation is to be specified, code the table using assembler DC statements as shown in the sample operation-level table "Operation-Level USS Table (ISTINCNO)" on page 785.

Table 88 shows the USS macroinstructions, their associated operands, and where the operands are described.

Table 88. USS Table Macroinstructions and Operands

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	USSTAB	DATEDLM	741		/	FORMAT; V4
		DATEFRM	741		MDY	FORMAT; DATEDLM; V4
		FORMAT	743		OLD	
		TABLE	751			
		TIMEDLM	752			FORMAT; V4

Table 88. USS Table Macroinstructions and Operands (continued)

Name	Macro instruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	USSCMD	CMD	740	R		
		FORMAT	743		PL1	
		REP	749			
name	USSPARM	PARM	748	R		
		DEFAULT	742			
		REP	749			
		VALUE	752			
name	USSMSG	BUFFER	739	R		Terminal operator; TEXT; OPT; SUPP
		MSG	745	R		
		TEXT	751	R		Required for terminal operator BUFFER
		DESC	742			VTAM operator
		MCSFLAG	745			VTAM operator
		MSGLVL	745		START	VTAM operator
		OPT	748		BLKSUP	terminal operator
					NOBLKSUP	VTAM operator
		ROUTCDE	750			VTAM operator
		SUPP	750		NEVER	terminal operator
name	USSEND					

Legend:

R Required operand.

Coding Notes

USSTAB Macroinstruction

The USSTAB macroinstruction indicates the beginning of a USS table.

USSCMD Macroinstruction

The USSCMD macroinstruction is used to define VTAM operator or terminal operator commands.

USSPARM Macroinstruction

The USSPARM macroinstruction defines an operand or positional parameter that can be specified on a command identified by the USSCMD macroinstruction. It also defines default values for the operand or positional parameter.

There can be multiple USSPARM macroinstructions associated with a USSCMD macroinstruction. For each operand (keyword and positional), code a USSPARM macroinstruction.

USSMSG Macroinstruction

The USSMSG macroinstruction defines VTAM operator messages (ISTxxxx) or terminal operator messages (USSMSGxx).

To determine under what circumstances these messages are issued, refer to *OS/390 SecureWay Communications Server: SNA Messages*.

USSEND Macroinstruction

The USSEND macroinstruction delimits the end of the USS table.

Full Syntax

The full syntax for the USS table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

USSTAB

```
►►──name──USSTAB──┬─DATEDLM=/──────────────┬──┬─DATEFRM=MDY──────────┬──►
                  └─DATEDLM=delimiter──────┘  └─DATEFRM=─┬─MDY─┬──────┘
                                                         ├─DMY─┤
                                                         └─YMD─┘

►──┬─FORMAT=DYNAMIC──────┬──┬──────────────┬──┬─TIMEDLM=:──────────┬──►◄
   │                     │  └─TABLE=name───┘  └─TIMEDLM=delimiter──┘
   └─FORMAT=─┬─DYNAMIC─┬──┘
             └─OLD─────┘
```

USSCMD

```
►►──┬──────┬──USSCMD──CMD=command_name──┬─FORMAT=PL1──────────┬──►
    └─name─┘                            └─FORMAT=─┬─PL1─┬─────┘
                                                  └─BAL─┘

►──┬──────────────────────────────────┬──►◄
   └─REP=replace_command_name─────────┘
```

USSPARM

```
►►──┬──────┬──USSPARM──PARM=──┬─operand_name─┬──┬──────────────────┬──►
    └─name─┘                  └─P_number─────┘  └─DEFAULT=value────┘
```

```
►──┬──────────────────┬──┬──────────────────┬──────────────────────────────────►◄
   └─REP=operand_name─┘  └─VALUE=value──────┘
```

USSMSG (Terminal Operator Messages)

```
►►──┬──────────┬──USSMSG──BUFFER=──┬─buffer_address──────────────┬──────────────►
    └─name─────┘                   ├─(buffer_address,LUNAME)─────┤
                                   └─(buffer_address,SCAN)───────┘

►─MSG=──┬─message_id──────────┬──TEXT='message_text'──┬─┬─OPT=BLKSUP──────────┬─┬──►
        └─(─message_id──────)─┘                       │ │                     │ │
                                                      └─└─OPT=──┬─BLKSUP──┬─┘ │
                                                                └─NOBLKSUP─┘   │

►──┬─┬─SUPP=NEVER──────┬─┬──────────────────────────────────────────────────►◄
   │ │                 │ │
   └─└─SUPP=──┬─NEVER──┬─┘
              └─ALWAYS─┘
```

USSMSG (VTAM Operator Messages)

```
►►──┬──────────┬──USSMSG──MSG=──┬─message_id──────────┬──TEXT='message_text'──────►
    └─name─────┘                └─(─message_id──────)─┘

►──┬──────────────────────────┬──┬──────────────────────────────────────────►
   └─DESC=(descriptor_code)───┘  └─MCSFLAG=(multiple_console_support_flag)─┘

►──┬─┬─MSGLVL=START────────┬─┬─┬─OPT=NOBLKSUP──────────┬─┬──────────────────►
   │ │                     │ │ │                       │ │
   └─└─MSGLVL=──┬─START──┬─┘ │ └─OPT=──┬─NOBLKSUP──┬─┘ │
                ├─BASE───┤   │         └─BLKSUP────┘   │
                ├─V4R1───┤   │                         │
                ├─V4R2───┤   │                         │
                ├─V4R3───┤   │                         │
                ├─V4R4───┤   │                         │
                └─CS390──┘   │                         │

►──┬──────────────────────────┬──┬──────────────────────────────────────────►◄
   └─ROUTCDE=(route_code)─────┘  └─SUPP=──┬─ALWAYS─┬──────────────────────┘
                                          ├─INFO───┤
                                          ├─NEVER──┤
                                          ├─NORM───┤
                                          ├─SER────┤
                                          └─WARN───┘
```

USSEND

```
►►──┬──────┬──USSEND──────────────────────────────────────────────────►◄
    └─name─┘
```

Operand Descriptions

The following section describes the operands you can code for the USS table.

BUFFER

```
►►──BUFFER=──┬─buffer_address──────────────┬──────────────────────────►◄
             ├─(buffer_address,LUNAME)─────┤
             └─(buffer_address,SCAN)───────┘
```

macroinstructions: *USSMSG*
dependencies: *TEXT, OPT, SUPP, terminal operator messages*

buffer_address
Specifies the address (name) of an area of storage defined to contain the message text and a header indicating the length of the message text. The storage area must be formatted as shown in Figure 6.

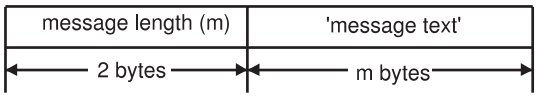


Figure 6. USS Message Layout in Storage

The message text defined in the storage area must follow the rules listed in “Syntax Rules for Messages” on page 754.

The message text is sent to the terminal operator as it appears in the storage area. VTAM does not modify or translate the message text (even if FEATUR2=LOWERCSE is specified for the device). You are responsible for including any device-dependent control characters within the message.

LUNAME|SCAN
Specifies that the character strings listed in Table 89 on page 740, will be replaced with the appropriate values in the position in the message where the character string occurred. The entire string specified by BUFFER will be searched, using the character @.

For example, if the following data was in the message text storage area

```
'@@LUNAME IS NOW IN SESSION WITH VTAM, @@@@DATE'
```

the message received at the terminal would look like the following:

```
'TERMA    IS NOW IN SESSION WITH VTAM, 01/04/94'
```

Note: For terminals with large screen sizes, searching the storage area for a character string might be a performance consideration.

Table 89. Valid Character Strings for Message Definition

Character String	In Message Text	Format
@@@@DATE	Current Date	In the format and with the delimiter specified in the DATEFRM and DATEDLM operands on the USSTAB macroinstruction.
@@LUNAME	Terminal name	The name is left-justified and trailing blanks are inserted, if the name is fewer than 8 characters.
@@@NETID	Network ID	The network ID is left-justified and trailing blanks are inserted, if the network ID is fewer than 8 characters. The network identifier for a nonnative-network-attached terminal is the real network name (the network identifier of the network where the terminal resides), not the name of the network it is attached to.
@@@@@@@@@@@@@@@NQN	Network-qualified name	The network-qualified name is left-justified and trailing blanks are inserted if the network-qualified name is fewer than 17 characters, as *networkID.name*.
@@@@RUNAME	Request/response unit	The name of the request or response unit is left-justified and trailing blanks are inserted, if the RU name is less than 10 characters. Applies only to USSMSG07 when coded using the BUFFER operand.
@@@SENSE	Sense code	Eight-character sense code. Applies only to USSMSG07 when coded using the BUFFER operand.
@@@@TIME	Current Time	As HH_MM_SS, where an underscore (_) is the delimiter specified on the TIMEDLM operand of the USSTAB macroinstruction.

Notes:

1. Although a message size of 65535 (X'FFFF') can be specified, ensure that the terminal can support the size of the message specified.
2. BUFFER and TEXT are mutually exclusive. VTAM produces an error message if both the BUFFER and TEXT operands are coded.
3. If you code either the OPT or SUPP operands in addition to BUFFER, the OPT or SUPP operand will be ignored.

CMD

```
►►──CMD=command_name──────────────────────────────────────────────────►◄
```

macroinstructions: *USSCMD*
dependencies: *none*

Specifies the user-defined command name to which this USSCMD macroinstruction applies. *command_name* can consist of from 1–8 characters, the first of which must be alphabetical (A–Z) or a national character (@, #, or $).

Note: If two commands are defined with the same *command_name*, the assembly of the table will fail.

DATEDLM

```
►►─┬─DATEDLM=/──────────────┬──────────────────────────────────────────►◄
   │                        │
   └─DATEDLM=delimiter──────┘
```

macroinstructions:	*USSTAB*
dependencies:	*FORMAT=DYNAMIC; V4 and later versions*

Specifies the character to be used as a delimiter to separate the month, day, and year parts of the date where @@@@DATE is specified in the message. The slash (/) is used if DATEDLM is not specified.

DATEDLM is valid only when FORMAT=DYNAMIC.

Note: An ampersand (&) and single quotation mark (') are not valid delimiters.

DATEFRM

```
►►─┬─DATEFRM=MDY────────────┬──────────────────────────────────────────►◄
   │                        │
   └─DATEFRM=─┬─MDY──┬──────┘
              │      │
              ├─DMY──┤
              │      │
              └─YMD──┘
```

macroinstructions:	*USSTAB*
dependencies:	*FORMAT=DYNAMIC; DATEDLM; V4 and later versions*

Specifies the date format to be used where @@@@DATE is specified in the message text. Note that the delimiter used between the month, day, and year is specified on the DATEDLM operand.

DATEFRM is valid only when FORMAT=DYNAMIC.

DATEFRM=DMY
Specifies the day, followed by month, followed by year as dd_mm_yy, where an underscore (_) is the delimiter specified on the DATEDLM operand.

DATEFRM=MDY
Specifies the month, followed by day, followed by year as mm_dd_yy, where an underscore (_) is the delimiter specified on the DATEDLM operand.

DATEFRM=YMD
Specifies the year, followed by month, followed by day as yy_mm_dd, where an underscore (_) is the delimiter specified on DATEDLM operand.

The DATEFRM operand overrides the DATEFORM start option. If the DATEFRM operand is not specified, the DATEFORM start option takes effect.

DEFAULT

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─DEFAULT=value────────┘
```

macroinstructions: *USSPARM*
dependencies: *VALUE*

Specifies a default value to be used if the operand is omitted when the command is entered. If DEFAULT is not specified, the operand is treated as if it were not entered.

If the parameter in the PARM operand allows a network-qualified name to be specified, then the value of DEFAULT can be a network-qualified name.

The DEFAULT and VALUE operands cannot be coded on the same USSPARM macroinstruction. Instead, code two USSPARM macroinstructions with the same value specified for PARM. The macroinstruction specifying VALUE must precede the one containing the DEFAULT operand. If REP is to be specified, it must be on the macroinstruction containing the VALUE operand.

The default value (*value*) must follow the rules for coding assembler DC statements for character (C-type) constants. For example, if you want the following character string for *value*

```
'DUMP'
```

code

```
DEFAULT='DUMP'
```

DESC

```
►►──┬──────────────────────┬──────────────────────────────────────────►◄
    └─DESC=(descriptor_code)─┘
```

macroinstructions: *USSMSG*
dependencies: *VTAM operator messages only*

Specifies the descriptor code to be used for this message. This operand is coded exactly like, and has the same meaning as, the DESC operand of the WTO or WTOR macroinstruction. For information on how to code this operand, refer to *MVS Supervisor Services and Macros*.

This operand applies only to messages that are sent to the VTAM operator. VTAM ignores this operand when sending a message to a program operator.

If the DESC operand is omitted or is specified with a null value (DESC=), VTAM uses the DESC value specified for this message ID in the next USS table searched.

FORMAT (USSTAB)

```
►►─┬─FORMAT=DYNAMIC──────────┬──────────────────────────────────────►◄
   │                         │
   │            ┌─DYNAMIC─┐  │
   └─FORMAT=────┤         ├──┘
                └─OLD─────┘
```

macroinstructions: *USSTAB*
dependencies: *none*

Specifies how the USS table will be formatted.

FORMAT=DYNAMIC
Specifies that the MODIFY TABLE command can be used to dynamically replace the USS table. Also, use FORMAT=DYNAMIC to permit terminal operators to specify the USS table using the LANGTAB operand on a USS command.

Specifying FORMAT=DYNAMIC can significantly reduce the CPU time required to search for VTAM operator messages. Refer to *OS/390 SecureWay Communications Server: SNA Operation* for more information on dynamically replacing the USS table.

FORMAT=DYNAMIC is coded in both the IBM-supplied USS tables — ISTINCNO and ISTINCDT.

FORMAT=OLD
Specifies that the table is not dynamically replaceable. This is the only value that can be specified for a pre-V3R2 system.

FORMAT (USSCMD)

```
►►─┬─FORMAT=PL1──────────────┬──────────────────────────────────────►◄
   │                         │
   │            ┌─PL1─┐      │
   └─FORMAT=────┤     ├──────┘
                └─BAL─┘
```

macroinstructions: *USSCMD*
dependencies: *none*

Specifies the syntax of the user-defined command to be entered by the terminal operator.

FORMAT=BAL
Specifies the user-defined command specified on this USSCMD macroinstruction is in Basic Assembler Language (BAL) syntax.

```
►►──command──┬──────────────
```

MCSFLAG

```
►►──┬──────────────
```

```
   ┌─MSGLVL=START──────┐
►►─┤                   ├──────────────────────────────────────────────►◄
   │           ┌─START─┐
   └─MSGLVL=───┤       ├
               ├─BASE──┤
               ├─V4R1──┤
               ├─V4R2──┤
               ├─V4R3──┤
               ├─V4R4──┤
               └─CS390─┘
```

macroinstructions:	*USSMSG*
dependencies:	*VTAM operator messages only*

Specifies the version of the message to issue on a message-by-message basis. MSGLVL is valid only for those messages listed in *OS/390 SecureWay Communications Server: SNA Messages* The value specified on the MSGLVL operand overrides the MSGLEVEL start option value for this message.

The V4R1 and above messages often use network-qualified names for session-capable resources to enable you to know specifically which resource is affected by the message. You can choose to have VTAM issue the message that uses non-network qualified names to the VTAM operator console, but be aware that the messages that use network-qualified names might be more descriptive. Since the MSGLEVEL start option has a default value of BASE, you do not have to do anything if you want VTAM to issue the pre-V4R1 messages.

For example, message

```
IST073I  command for ID=nodename FAILED - MORE POWERFUL REQUEST IN PROGRESS
```

has been replaced by the following message group

```
IST1129I  command FAILED, nodename - DEACTIVATE PENDING
IST1045I  NODE TYPE = nodetype
IST314I   END
```

If you want to continue to get IST073I, code the following:

```
[name]   USSMSG     MSG=IST1129I,MSGLVL=BASE
```

MSGLVL=BASE

Specifies that for the message number indicated on the MSG operand, issue the pre-V4R1 message that this message number replaces, regardless of the value coded on the MSGLEVEL start option.

For messages replaced by message groups, to get the base message, code MSGLVL=BASE for the first message in the group. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages supported by the MSGLVL operand.

Note: If you specify MSGLVL=BASE and if there are duplicate names in your SNA interconnected networks, you might not know which resource the message applies to.

MSGLVL=START

Specifies that for the message number indicated on the MSG operand, issue the version of the message specified on the MSGLEVEL start option. START is the default for only those messages supported by the MSGLVL operand.

MSGLVL=V4R1
Specifies that for the message number indicated on the MSG operand, issue the V4R1 message or messages, regardless of the value specified on the MSGLEVEL start option. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of V4R1 messages supported by the MSGLVL operand.

MSGLVL=V4R2
Specifies that for the message number indicated on the MSG operand, issue the V4R2 message or messages, regardless of the MSGLVL value specified on the MSGLEVEL start option. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages supported by the MSGLVL operand.

MSGLVL=V4R3
Specifies that for the message number indicated on the MSG operand, issue the V4R2 message or messages, regardless of the MSGLVL value specified on the MSGLEVEL start option. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages supported by the MSGLVL operand.

MSGLVL=V4R4
Specifies that for the message number indicated on the MSG operand, issue the V4R2 message or messages, regardless of the MSGLVL value specified on the MSGLEVEL start option. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages supported by the MSGLVL operand.

MSGLVL=CS390
Specifies that for the message number indicated on the MSG operand, issue the CS for OS/390 message or messages, regardless of the MSGLVL value specified on the MSGLEVEL start option. Refer to *OS/390 SecureWay Communications Server: SNA Messages* for a list of messages supported by the MSGLVL operand.

name

```
►►──name──────────────────────────────►◄   ►►──┬──────┬──────────────────────►◄
                                              └name──┘
```

macroinstruction:	USSTAB	*macroinstruction:*	USSCMD; USSPARM; USSMSG; USSEND

For USSCMD, USSPARM, USSMSG, USSEND: Specifies the name assigned to the macroinstruction.

For USSTAB: Specifies the required CSECT name for the USS table.

OPT

For Terminal Operator

```
►►─┬─OPT=BLKSUP──────────────┬──────────────────────────►◄
   │                         │
   └─OPT=─┬─BLKSUP──┬─────────┘
          └─NOBLKSUP─┘
```

For VTAM Operator

```
►►─┬─OPT=NOBLKSUP──────────────┬──────────────────────────►◄
   │                           │
   │                           │
   └─OPT=─┬─NOBLKSUP──┬─────────┘
          └─BLKSUP────┘
```

macroinstructions:	USSMSG	*macroinstructions:*	USSMSG
dependency:	BUFFER	*dependency:*	none

Indicates if extraneous blanks are to be suppressed from the defined message.

OPT=BLKSUP
Specifies that extraneous blanks are to be suppressed from the message. Any sequence of two or more blanks is to be converted into a single blank.

OPT=NOBLKSUP
Specifies that extraneous blanks are not to be suppressed from the message. Any sequence of two or more blanks is to be presented unchanged in the message.

Notes:

1. For terminal operators VTAM ignores the OPT operand, if both OPT and BUFFER are specified.
2. It is recommended that OPT=NOBLKSUP be used for any message that is part of a tabular display.
3. If the OPT operand is omitted or is specified with a null value (OPT=), VTAM uses the OPT value specified for this message ID in the next USS table searched.

PARM

```
►►─PARM=─┬─operand_name─┬──────────────────────────────────►◄
         └─P_number─────┘
```

macroinstructions:	*USSPARM*
dependencies:	*none*

Specifies a user-defined keyword or positional parameter that can be coded in the user-defined command identified by the USSCMD definition.

PARM=*operand_name*
Specifies the keyword parameter in the user-entered command to which this USSPARM macroinstruction applies. *operand_name* must be 1–8 alphanumeric characters.

PARM=*P_number*
Specifies a positional parameter, where *number* is a decimal integer from 1 to the maximum number of positional parameters for the command. *P_number* indicates the positional parameter in the user-entered command to which this USSPARM macroinstruction applies.

See Example 2 in “Appendix B. Examples of USS Command Conversion” on page 833 for an example of positional and keyword parameters.

REP (USSCMD)

```
►►──┬──────────────────────────┬──────────────────────────────────────────►◄
    └REP=replace_command_name──┘
```

macroinstructions: *USSCMD*
dependencies: *none*

Specifies the valid command that is to replace the user-defined command specified by the CMD operand. See Example 2 in “Appendix B. Examples of USS Command Conversion” on page 833 for an example of a user-defined command that is to be replaced.

If the REP operand is not coded, the value specified in the CMD operand is used. In this case, CMD must specify a valid command. See Example 5 in “Appendix B. Examples of USS Command Conversion” on page 833 for an example.

REP (USSPARM)

```
►►──┬──────────────────┬──────────────────────────────────────────────────►◄
    └REP=operand_name──┘
```

macroinstructions: *USSPARM*
dependencies: *none*

REP=*operand_name*
Specifies the parameter is replaced with *operand_name*. The value for *operand_name* must be 1–8 alphanumeric characters. The value of the operand is assigned from the parameter specified by PARM. If PARM specifies an keyword parameter, its value is assigned to the operand specified by REP. If PARM specifies a positional parameter, its value is treated as if it were an operand value and it is assigned to the operand specified by REP.

If REP is not coded, it takes the value of PARM. (That is, the user-entered parameter is used as entered.)

Positional parameters such as P1 and P2 can also be used as operands. For multiple specifications of the same parameter, the last value specified is used (as shown in Example 3 in “Appendix B. Examples of USS Command Conversion” on page 833).

REP=VERB
Indicates that the command is replaced with the value of the VALUE operand. The USS table is then searched a second time, matching this new command to the value of a CMD operand on a USSCMD macroinstruction.

For example using the following USS macroinstructions, if the operator enters MODIFY DUMP, during a second pass of the USS table VTAM will look for USSCMD CMD=DUMP.

```
USSCMD CMD=MODIFY
USSPARM PARM=DUMP,REP=VERB,VALUE=DUMP

USSCMD CMD=DUMP
USSPARM PARM=DUMPDS
```

ROUTCDE

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─ROUTCDE=(route_code)─────────┘
```

macroinstructions: *USSMSG*
dependencies: *VTAM operator messages only*

Specifies the routing code to be used for this message. This operand is coded exactly like, and has the same meaning as, the ROUTCDE operand of the WTO or WTOR macroinstruction. For information on how to specify this operand, refer to *MVS Supervisor Services and Macros*.

This operand applies only to messages that are sent to the VTAM operator. VTAM ignores this operand when sending a message to a program operator.

If the ROUTCDE operand is omitted or is specified with a null value (ROUTCDE=), VTAM uses the ROUTCDE value specified for this message ID in the next USS table searched.

SUPP

Terminal Operator

```
      ┌─SUPP=NEVER──────────┐
►►────┼─────────────────────┼──────────────────────────────────────►◄
      │          ┌─NEVER──┐ │
      └─SUPP=────┼────────┼─┘
                 └─ALWAYS─┘
```

macroinstruction: USSMSG
dependency: BUFFER

VTAM Operator

```
►►──┬──────────────────────────────┬──►◄
    └─SUPP=────────────────────────┘
           ┌─ALWAYS─┐
           ├─INFO───┤
           ├─NEVER──┤
           ├─NORM───┤
           ├─SER────┤
           └─WARN───┘
```

macroinstruction: USSMSG
dependency: none

Specifies the message suppression class to be used for this message. The message suppression class for a message, along with the SUPP start option and the MODIFY SUPP command, determines whether a particular message will be suppressed by VTAM.

If the SUPP operand is omitted or is specified with a null value (SUPP=), VTAM uses the SUPP value specified for this message ID in the next USS table searched.

You cannot suppress individual VTAM messages in a multi-line WTO (MLWTO) message (that is, messages identified as issued as part of a message group). You can suppress message groups by suppressing the message identified as the first in the group. Refer to *OS/390 SecureWay Communications Server: SNA Messages* to identify messages issued as part of a group.

SUPP=ALWAYS
Specifies that this message is always to be suppressed, regardless of what level of message suppression is in effect.

SUPP=INFO
Specifies that this message is to be in the informational message suppression class.

SUPP=NEVER
Specifies that this message is never to be suppressed, regardless of what level of message suppression is in effect. That is, this message should always be written.

SUPP=NORM
Specifies that this message is to be in the normal message suppression class.

SUPP=SER
Specifies that this message is to be in the serious message suppression class.

SUPP=WARN
Specifies that this message is to be in the warning message suppression class.

For terminal operators::

- INFO, NORM, SER, or WARNING may also be coded for SUPP, but each of these is processed as though SUPP=NEVER were specified.
- If you code both the SUPP and BUFFER operands, VTAM ignores SUPP.

TABLE

```
►►──┬──────────────┬──────────────────────────────────────────────────►◄
    └─TABLE=name───┘
```

macroinstructions:	*USSTAB*
dependencies:	*none*

Specifies a translation table to be used by VTAM to translate character-coded commands. If the table is not part of the module containing USSTAB, an EXTRN statement must be coded for the specified name.

If no translation table is specified, VTAM uses the translation table associated with the IBM-supplied USS table (or its user-written replacement). If the IBM-supplied table (or its user-written replacement) does not have a translation table, VTAM does no character translation.

TEXT

```
►►──TEXT='message_text'──────────────────────────────────────────────►◄
```

macroinstructions:	*USSMSG*
dependencies:	*BUFFER (terminal operator)*

Specifies the text to replace the USS messages identified by the MSG operand. The message text provided here must follow the rules listed in “Syntax Rules for Messages” on page 754.

Note: For a terminal operator message, VTAM produces an error message if you code both the BUFFER and TEXT operands.

On TEXT, you can place any combination of the character strings described in Table 89 on page 740; VTAM will replace the strings with the values shown in the table.

For example, the message definition

```
TEXT='@@LUNAME IS NOW IN SESSION WITH VTAM'
```

produces the following message:

```
TERMA     IS NOW IN SESSION WITH VTAM
```

Note: The message appears as shown above, even if OPT=BLKSUP is coded.

If the LUNAME has 8 characters, it produces a message like the following:

```
TERMBBBB IS NOW IN SESSION WITH VTAM
```

An example using date and time follows:

```
TEXT='IT IS @@@@TIME ON @@@@DATE'
```

produces a message like the following:

```
IT IS 10:14:51 ON 10/10/91
```

TIMEDLM

```
►►─┬─TIMEDLM=:──────────┬──────────────────────────────────────────►◄
   │                    │
   └─TIMEDLM=delimiter──┘
```

macroinstructions: *USSTAB*
dependencies: *FORMAT=DYNAMIC; V4 and later versions*

Specifies the character to be used as a delimiter to separate the hour, minutes, and seconds parts of the time. The colon (:) is used if TIMEDLM is not specified. The ampersand (&) and single quote (') are not valid.

TIMEDLM is valid only when FORMAT=DYNAMIC, and is not supported by pre-V4 VTAMs.

VALUE

```
►►──┬──────────────┬──────────────────────────────────────────────►◄
    └─VALUE=value──┘
```

macroinstructions: *USSPARM*
dependencies: *DEFAULT*

Specifies the default value to be used if the operand specified by the PARM operand is entered without a value (see Example 3 in “Appendix B. Examples of USS Command Conversion” on page 833).

VALUE is in contrast with the DEFAULT operand, which specifies the default to be used if the operand itself is not entered.

Notes:

1. If multiple VALUE operands are specified for the same operand, the first VALUE operand is used.
2. If the parameter in the PARM operand allows a network-qualified name to be specified, then the value of VALUE can be a network-qualified name.
3. The DEFAULT and VALUE operands cannot be coded on the same USSPARM macroinstruction. To use both operands, code two USSPARM macroinstructions with the same value specified for PARM. The macroinstruction specifying VALUE must precede the one containing the DEFAULT operand. If REP is to be specified, it must be on the macroinstruction containing the VALUE operand. For example,

```
USSPARM P=T,REP=TYPE,VALUE=COND
USSPARM P=T,REP=TYPE,DEFAULT=COND
```

Customizing Messages

VTAM provides an extensive set of messages that provide information to a terminal operator, to a program operator, or to a VTAM operator. You can redefine any of these messages by coding a USSMSG macroinstruction for each message to be redefined. These USSMSG macroinstructions are included in a user-defined USS table that is associated with the terminal, a program operator application, or the system console (for messages to the VTAM operator).

Using the USSMSG macroinstruction, you can change the message text and suppression class of any message.

Notes:

1. For information on changing message characteristics, refer to *OS/390 SecureWay Communications Server: SNA Messages*.
2. For more information on how you can use the MVS Message Service to change the text of USS messages received by terminal users, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.
3. Because program operator applications depend on VTAM messages, changes to the operator messages could disrupt the functioning of a program operator applications.

A message might be only partially defined in a USS table. This allows options on messages to be received from a USS table that is next in the search order hierarchy. See “Order of USS Table Use” on page 733 for more information on the search order for USS tables.

More than one message can be defined using a single USSMSG macroinstruction. If more than one message ID is specified, all of the corresponding messages are changed by the operands specified with this macroinstruction.

The USSMSGxx messages have VTAM equivalents defined in the IBM-supplied USS tables. For example, ISTINCNO specifies

```
USSMSG MSG=(IST455I,11)
```

to define VTAM message IST455I and USSMSG11. These messages can be redefined using a single USSMSG macroinstruction.

Table 90 shows the VTAM message equivalent of USSMSG00 to USSMSG14.

Table 90. USS Messages and VTAM Message Equivalents

USS Message Number	VTAM Operator Message
0	IST457I
1	IST450I
2	IST451I
3	IST452I
4	IST453I
6	IST792I
8	IST454I
11	IST455I
12	IST456I
14	IST458I

Note: Allowing the default USSMSG10 to be sent to a non-SNA printer could cause printer forms to be out of alignment. You can avoid this by coding a separate USSTAB for these printers and coding USSMSG10 in this table as follows:

```
USSMSG MSG=10,BUFFER=BUF10
BUF10     DC    X'0003F57A19'
```

Syntax Rules for Messages

When preparing USS messages to be sent to a terminal operator, to a VTAM operator, or to a program operator, the following rules apply:

- Single quotation marks in the message text must be specified as in the assembler DC statement for character (C-type) constants.
- The message text must conform to the rules for coding an assembler DC statement for character (C-type) constants. Alternatively, if noncharacter message text is required, the TEXT operand can be specified as a sublist, with each entry coded as the full operand of an assembler DC statement. If the sublist form is used, the assembler does boundary alignment processing. For example,

  ```
  TEXT=(A(0),C'SAMPLE MESSAGE')
  ```

  might result in some leading zeroes being supplied by the assembler to ensure that the “A(0)” part of the text is located on a fullword boundary. This could be avoided by specifying

  ```
  TEXT=(AL4(0),C'SAMPLE MESSAGE')
  ```
- If the message text contains blanks (X'40'), OPT=BLKSUP should not be specified for this message, because VTAM might delete some of the X'40' characters (unless that is desired).
- You cannot use the 3270 control commands and orders when creating USS messages for SNA devices. For further information, see the component description manual for the appropriate control unit and attached devices.
- When VTAM issues a message, it can supply one or more variable data character strings to be inserted in the message text. The location of the variable data character strings in the message text is indicated by numbered or unnumbered percent signs (%).

– Numbered percent signs are used when a message has more than one variable character string. The first variable data string is represented by %(1), the second string by %(2), and so on, for as many strings as are provided by VTAM.
– An unnumbered percent sign is the same as specifying %(1).

You can rearrange the order of the percent signs in a message to change the order of the variable data in a message.

For example, if the message definition for IST380I:

```
TEXT='ERROR FOR ID = %(1) – REQUEST: %(2), SENSE: %(3)'
```

results in the message

```
ERROR FOR ID = NODE1234 – REQUEST: CDINIT, SENSE: 087D0001
```

Then this message definition for IST380I:

```
TEXT='%(2) REQUEST RECEIVED SENSE %(3) FOR NODE %'
```

results in the message

```
CDINIT REQUEST RECEIVED SENSE 087D0001 FOR NODE NODE1234
```

Note: Noncharacter data should not contain percent signs (X'6C') because VTAM interprets a percent sign as a point at which to insert variable data into the message.

- Variable text can be used more than once in the same message or can be omitted in the redefined message.

  ```
  TEXT='ERROR FOR ID = %(1) – REQUEST: %(2), SENSE: %(3)–%(2)'
  ```

  would result in the message

  ```
  ERROR FOR ID = NODE1234 – REQUEST: CDINIT, SENSE: 087D0001–CDINIT
  ```

- If the number of a percent sign is greater than the number of variable data character strings provided for that message, that percent sign is replaced by the last available variable data.

  For example, if %(7) were specified for the following message, the values for %(3) will be used.

  ```
  TEXT='ERROR FOR ID = %(1) – REQUEST: %(2), SENSE: %(3)'
  ```

  OS/390 SecureWay Communications Server: SNA Messages shows the number and location of the variable data character strings in each VTAM message. It also provides a message explanation that includes a description of what type of data can appear in each character string.

- The maximum length of a message after replacement of any percent signs is 251 characters. If a message exceeds 251 characters, the USS table will not assemble. The maximum limit for messages in a message group is 60 characters for each message. Message will be truncated if maximum is exceeded.
- Do not change the message ID that appears in the text of a VTAM operator message. If you do, you could affect IBM's ability to service your product.
- The following characters can be used for USS messages:
  – 26 uppercase and lowercase letters: A–Z, a–z
  – 3 national characters: $ # @
  – 10 numerical digits: 0–9
  – All graphic characters great than or equal to X'40' (with the exception of X'6C' which is the percent sign)

National characters (and any graphic or control characters not listed above) are sent to a terminal user only if present in user-specified message replacements specified through the BUFFER operand of the USSMSG macroinstruction.

Note: (For terminal operator messages only.) Because control characters are device-dependent, you should not include them in the data defined by the TEXT or BUFFER operands, unless you can select the appropriate character for the device to which the message will be sent. You may use the percent sign and blanks in data defined by the BUFFER operand. However, do not include the percent sign or blanks in the TEXT operand data because USS treats them differently.

Defining VTAM Operator Commands

VTAM operator commands use the Basic Assembler Language (BAL) format. You have the option of specifying PL/I format for all USS commands in your USS table. This is not recommended, however, because using different formats for the VTAM commands might be confusing.

Unlike messages, a command must be completely defined in a single USS table.

Syntax Rules for VTAM Operator Commands

The first two parts of each command, the command name and the first operand cannot be changed by the USS table.

Command	1st Operand
DISPLAY	NET
MODIFY	procname
VARY	NET

All other operands of a VTAM command can be changed in a user-defined table.

Note: The VTAM START and HALT commands cannot be changed through a USS table.

Include definitions for all operands, even if you are not redefining all of them. Any operand that does not appear in the user table will be passed unconverted to the command processor.

- Operands that describe the command function (such as TERM on the VARY NET,TERM command, ADJSSCPS on the DISPLAY NET,ADJSSCPS) cannot be changed. However, other operands can be redefined (such as LU1, LU2, and NOTIFY on the VARY NET,TERM command).

  ```
  VARY NET,TERM,LU1=luname,LU2=lu2name,NOTIFY=YES
  ```

- Keyword operands can be converted into another operand with the same value. For example

  ```
  MODIFY MSG=YES
  ```

  can be converted into

  ```
  MODIFY MSGMOD=YES
  ```

- It is also possible to specify a default value to be used if the operand is omitted or specified without a value.

  ```
  USSPARM PARM=MSGMOD,VALUE=YES
  USSPARM PARM=MSGMOD,DEFAULT=YES
  ```

```
MODIFY procname,MSGMOD=,
MODIFY procname
```

- An operand value specified by the operator cannot be replaced using the USS table.

  For example,

  ```
  MODIFY procname,MSGMOD=NO
  ```

  cannot be converted into

  ```
  MODIFY procname,MSGMOD=YES
  ```

- Commands cannot be redefined to perform more than one function. For example, the following commands

  ```
  DISPLAY NET,CDRMS
  DISPLAY NET,CLSTRS
  ```

  cannot be converted to one command

  ```
  DISPLAY NET,CDRMS,CLSTRS
  ```

Defining Terminal Operator Commands

VTAM recognizes the following terminal operator commands that originate from a terminal operator. These terminal operator commands are received from a logical unit that sends unformatted character-coded logon and logoff requests:

- “LOGON Command” on page 758
- “LOGOFF Command” on page 762
- “IBMTEST Command” on page 765

VTAM converts each of these unformatted commands into a field-formatted SNA request.

Syntax Rules for Terminal Operator Commands

The following syntax rules apply to commands you define for terminal operator commands. If these rules are not followed, use an interpret table to convert the character-coded command into a formatted SNA request.

- The following characters are allowed:
  - All graphics characters (greater than or equal to X'40')
  - BS (backspace: X'16')
  - HT (horizontal tab: X'05')
  - IRS (interchange record separator: X'1E')
  - NL (new line: X'15')

  Note: New line characters are deleted from the character string before translation if you specify SSCPFM=FSS or SSCPFM=USSSCS on the LU definition statement.

- Command names and operand values that identify a name can contain 1–8 characters in the following format:

1st character	alphabetical (upper or lowercase) or the national characters @, #, or $
2nd to 8th characters	alphanumeric or the national characters @, #, or $

- Command names cannot contain commas except between paired single quotation marks or parentheses.

- Commands names in the BAL format cannot contain blanks or horizontal tab characters except between the command and first operand or between paired single quotation marks.
- For PL/1 format, operand values cannot contain semicolons except between paired single quotation marks Positional operand values cannot contain commas except between paired single quotation marks.
- Operand values cannot contain blanks, horizontal tabs, or unpaired parentheses except between paired single quotation marks.

  Note: There cannot be an odd number of single quotation marks.
- For BAL commands, the terminal operator cannot use the following for operand values:
  - Commas except between paired single quotation marks or parentheses
  - A positional parameter cannot contain equal signs except between paired parentheses or single quotation marks
- Operand values can contain any of the following characters:
  - All graphics characters (greater than or equal to X'40')
  - BS (backspace: X'16')
  - HT (horizontal tab: X'05')
  - Data entered from a magnetic card reader

  Note: Magnetic card data from a BSC or type 1 PU 3270 device is supported only if the card data is used as the last data in a value within quotation marks for the last operand of a USS command. Press the CLEAR key before you enter a character-coded command that contains magnetic card reader data.

LOGON Command

The LOGON command allows the terminal operator to request a session with an application program.

Table 91. LOGON Command and Operands

Command	Operands	Page	Req	Default	Restrictions, Dependencies
LOGON	APPLID(*appl_prog_name* or *uservar_name*)	759	R		
	DATA(*user_data*)	759			
	LANG(*language_code* or *language_name*)	759			
	LANGTAB(*uss_table_name*,*language_code*)	760			
	LOGMODE(*logon_mode_entry*)	761			
	MODEL(*model_name*)	761			
	PRINTER1(*primary_printer_name*)	761			
	PRINTER2(*alternate_printer_name*)	761			

Legend:

R Required operand.

Full Syntax

The full syntax for the LOGON command follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

```
►►──LOGON──APPLID──┬─(appl_prog_name)─┬──┬──────────────────────────►
                   └─(uservar_name)───┘  └─DATA(user_data)──┘

►──┬──────────────────────────┬──┬──────────────────────────────────►
   └─LANG──┬─(language_code)─┬─┘  └─LANGTAB(USS_table_name,language_code)─┘
           └─(language_name)─┘

►──┬──────────────────────────┬──┬──────────────────────────────────►
   └─LOGMODE(logon_mode_entry)─┘  └─MODEL(model_name)─┘

►──┬──────────────────────────────────────────┬──┬──────────────────────────────────────────┬──►◄
   └─PRINTER1(primary_printer_name)─┘  └─PRINTER2(alternate_printer_name)─┘
```

Operand Descriptions

The following section describes the operands you can code for the LOGON command.

APPLID

```
►►──APPLID──┬─(appl_prog_name)─┬──────────────────────────────────►◄
            └─(uservar_name)───┘
```

Specifies the name of the application program or character string specified by the SEQNCE entry in the interpret table (USERVAR) with which a session is to be established. For more information on network regulations regarding name specification, see “APPL Definition Statement” on page 29.

DATA

```
►►──┬──────────────────────────┬──────────────────────────────────►◄
    └─DATA(user_data)──┘
```

Specifies user data to be made available to the application program’s logon exit routine. To embed blanks, code *user_data* as a single-quoted string. The application receiving the user data will not receive the quotes.

LANG

```
►►──┬──────────────────────────┬──────────────────────────────────►◄
    └─LANG──┬─(language_code)─┬─┘
            └─(language_name)─┘
```

dependencies: *MVS Message Service active*

Specifies the language in which the terminal operator will receive USS messages from the MVS Message Service. You are responsible for defining these USS

messages to the MVS Message Service and for having the MVS Message Service active in the host. Otherwise, the terminal operators will not receive USS messages from the MVS Message Service in the language of their choice. The terminal user continues to receive USS messages in the specified language, until the terminal user enters another valid USS command using the LANG operand.

For the LOGON command only, the specified language is also passed to the application program during logon processing.

If you enter this operand without a *language_code* or *language_name* specified the MVS Message Service is not used for USS messages.

Note: If the MVS Message Service is not currently active, the LANG operand is ignored.

LANG(*language_code***)**
Specifies a 3-character code that indicates a valid language. For a list of valid language code values, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

(LANG=*language_name***)**
Specifies the character string associated with a language. You can use the MVS Message Service to define your own values for the *language_name*.

For information on the MVS Message Service, refer to *MVS/ESA Planning: Operations* and the *MVS/ESA Application Development* Guide: Assembler Language Programs*.

LANGTAB

```
►►──┬──────────────────────────────────────────────┬──────────────────────────►◄
    └─LANGTAB(USS_table_name,language_code)─────────┘
```

Specifies the session-level USS table that VTAM uses for USS messages and commands. It also specifies the language that is passed to the application program. This operand provides the same function as the LANG operand, except it does not use the MVS Message Service.

USS_table_name
Specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

language_code
Specifies a 3-character code used to specify the language that is passed to the application program during logon processing. If a valid LANG operand is also specified and the MVS Message Service is active, this parameter is overridden by the language specified with the LANG operand. For a list of valid values and for details on how the application obtains this information, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

Notes:

1. If you enter this operand without a USS table name, it removes the LANGTAB USS table (specified earlier) from the search hierarchy.

2. If you enter this operand without a language code, the application receives the language from the LANG operand of the MODEENT macroinstruction (specified by LOGMODE).

LOGMODE

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─LOGMODE(logon_mode_entry)────┘
```

Specifies the logon mode entry that is used to select a set of session parameters for the session to be established.

MODEL

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─MODEL(model_name)────────────┘
```

Specifies the 1–8 character model name to be passed to the application program during logon processing.

If this operand is entered with a name, the data it contains is used. Table specifications on the terminal's definition statement are ignored for that type of data during the current session.

If this operand is entered without a name value, an override with null data results, and no model name data is sent to the application program for the requested session.

PRINTER1

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─PRINTER1(primary_printer_name)─┘
```

Specifies the primary printer name to be passed to the application program during logon processing.

If this operand is entered with a name, the data it contains is used. Table specifications on the terminal's definition statement are ignored for that type of data during the current session.

If this operand is entered without a name value, an override with null data results, and no primary printer name data is sent to the application program for the requested session.

PRINTER2

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─PRINTER2(alternate_printer_name)─┘
```

Specifies the alternate printer name to be passed to the application program during logon processing.

If this operand is entered with a name, the data it contains is used. Table specifications on the terminal's definition statement are ignored for that type of data during the current session.

If this operand is entered without a name value, an override with null data results, and no alternate printer name data is sent to the application program for the requested session.

LOGOFF Command

The LOGOFF command allows the terminal operator to request that a session with an application program be ended.

Table 92. LOGOFF Command and Operands

Command	Operands	Page	Req	Default	Restrictions, Dependencies
LOGOFF	APPLID(*appl_prog_name*)	763			
	HOLD(*YES* or *NO*)	763		YES	token ring
	LANG(*language_code* or *language_name*)	763			MVS Message Service
	LANGTAB(*uss_table_name*,*language_code*)	764			
	TYPE(COND or FORCE or UNCOND)	764		UNCOND	

Full Syntax

The full syntax for the LOGOFF command follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

```
                                          ┌─HOLD(YES)──────────┐
►►─LOGOFF──┬──────────────────────────────┼────────────────────┼──────────────►
           └─APPLID(appl_prog_name)──────┘                    │
                                          └─HOLD(─┬─YES─┬─)──┘
                                                  └─NO──┘

►──┬──────────────────────────────┬──┬──────────────────────────────────┬──►
   └─LANG──┬─(language_code)──┬──┘  └─LANGTAB(USS_table_name,language_code)─┘
           └─(language_name)──┘

   ┌─TYPE(UNCOND)──────────┐
►──┼───────────────────────┼──────────────────────────────────────────────►◄
   │                       │
   └─TYPE(─┬─UNCOND─┬─)──┘
           │─COND───│
           └─FORCE──┘
```

Operand Descriptions

The following section describes the operands you can code for the LOGOFF command.

APPLID

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─APPLID(appl_prog_name)───────┘
```

Specifies the name of the application program with which a session is to be terminated. If omitted, the application program with which a current session exists is assumed.

HOLD

```
    ┌─HOLD(YES)──────────┐
►►──┼────────────────────┼──────────────────────────────────────────►◄
    │          ┌─YES─┐   │
    └─HOLD(────┼─────┼───┘
               └─NO──┘)
```

dependencies: *do not code for token ring*

Specifies the action the logical unit expects VTAM to take when physically terminating the SSCP-PU session after the logical unit has been disconnected. A HOLD value of YES corresponds to a NOT LAST indicator on a Terminate Self request. The effect of HOLD depends on the setting of the DISCNT operand specified on the PU statement. The relationship between HOLD and DISCNT is described in the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Note: HOLD(NO) should not be specified for token ring connections as this might cause errors in attached gateways which require operator intervention.

LANG

```
►►──┬──────────────────────────────┬──────────────────────────────────►◄
    └─LANG──┬─(language_code)──────┬─┘
            └─(language_name)──────┘
```

dependencies: *MVS Message Service active*

Specifies the language in which the terminal operator will receive USS messages from the MVS Message Service. You are responsible for defining these USS messages to the MVS Message Service and for having the MVS Message Service active in the host. Otherwise, the terminal operators will not receive USS messages from the MVS Message Service in the language of their choice. The terminal user continues to receive USS messages in the specified language, until the terminal user enters another valid USS command using the LANG operand.

If you enter this operand without a language code or name, USS messages are no longer retrieved from the MVS Message Service.

Note: If the MVS Message Service is not currently active, the LANG operand is ignored.

language_name
Specifies the character string associated with a language. You can use the MVS Message Service to define your own values for the *language_name*.

language_code
Specifies a 3-character code that indicates a valid language. For a list of valid language code values, see *OS/390 eNetwork Communications Server: SNA Programming*.

For information on the MVS Message Service, see *MVS/ESA Planning: Operations* and the *MVS/ESA Application Development Guide: Assembler Language Programs*.

LANGTAB

```
►►──┬──────────────────────────────────────────┬──────────────────────────────►◄
    └─LANGTAB(USS_table_name,language_code)─────┘
```

Specifies the session-level USS table that VTAM uses for USS messages and commands. It also specifies the language that is passed to the application program. This operand provides the same function as the LANG operand, except it does not use the MVS Message Service.

USS_table_name
Specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

language_code
Specifies a 3-character code used to specify the language that is passed to the application program during logon processing. If a valid LANG operand is also specified and the MVS Message Service is active, this parameter is overridden by the language specified with the LANG operand. For a list of valid values and for details on how the application obtains this information, see *OS/390 eNetwork Communications Server: SNA Programming*.

Notes:

1. If you enter this operand without a USS table name, it removes the LANGTAB USS table (specified earlier) from the search hierarchy.
2. If you enter this operand without a language code, the application receives the language from the LANG operand of the MODEENT macroinstruction (specified by LOGMODE).

TYPE

```
    ┌─TYPE(UNCOND)──────────┐
►►──┼───────────────────────┼──────────────────────────────────────────────►◄
    │          ┌─UNCOND─┐   │
    └─TYPE(────┼────────┼──)┘
               ├─COND───┤
               └─FORCE──┘
```

Specifies the manner in which an active session between an application acting as a primary logical unit (PLU) and another logical unit is to be ended.

TYPE(COND)
Specifies that the LOSTERM exit routine of the PLU receives a code indicating conditional session ending.

TYPE(FORCE)
Specifies that the PLU's NSEXIT exit routine receives a code indicating forced ending and the session is brought down immediately by the SSCP. If there is no NSEXIT exit routine, VTAM sends the code to the LOSTERM exit routine, which must issue a CLSDST macroinstruction.

TYPE(UNCOND)
Specifies that the PLU's LOSTERM exit is driven with a code indicating unconditional session ending. SENDs and RECEIVEs stop immediately, and the PLU is to clean up storage used for the session by issuing a CLSDST macroinstruction.

UNCOND is used for normal situations when the session is assumed to be viable and an UNBIND can be sent to end the session. FORCE is used in abnormal situations, for example, when either one of the session ends cannot communicate with the SSCP. When use of the UNCOND operand has no effect, the FORCE operand should be specified.

IBMTEST Command

The IBMTEST command allows the terminal operator to test the physical path between the terminal and VTAM. IBMTEST causes VTAM to return test data to the terminal a specified number of times. VTAM returns the test data with the prefix "IBMECHO".

Table 93. IBMTEST Command and Operands

Command	Operands	Page	Req	Default	Restrictions, Dependencies
IBMTEST	*number*	767	R	10	
	LANG(*language_code* or *language_name*)	766			
	LANGTAB(*uss_table_name*,*language_code*)	766			
	test_data	767			

Legend:

R Required operand.

Full Syntax

The full syntax for the IBMTEST command follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

```
►►──IBMTEST──┬─10─────┬──┬──────────────────────────────────────────►
             └─number─┘  │                                          │
                         └─LANG=──┬─language_code─┬──┘
                                  └─language_name─┘

►──┬──────────────────────────────────────────┬──┬──────────────┬──►◄
   └─LANGTAB(USS_table_name,language_code)─┘  └─test_data─┘
```

Operand Descriptions

The following section describes the operands you can code for the IBMTEST command.

LANG

```
►►──┬──────────────────────────────────────────────────────────────────────────►◄
    └LANG=──┬─language_code─┬──┘
            └─language_name─┘
```

Specifies the language in which the terminal operator will receive USS messages from the MVS Message Service. You are responsible for defining these USS messages to the MVS Message Service and for having the MVS Message Service active in the host. Otherwise, the terminal operators will not receive USS messages from the MVS Message Service in the language of their choice. The terminal user continues to receive USS messages in the specified language, until the terminal operator enters another valid USS command using the LANG operand.

If you enter this operand without a *language_code* or *language_name*, USS messages are no longer retrieved from the MVS Message Service.

Note: If the MVS Message Service is not currently active, the LANG operand is ignored.

language_name
Specifies the character string associated with a language. You can use the MVS Message Service to define your own values for the *language_name*.

language_code
Specifies a 3-character code that indicates a valid language. For a list of valid language code values, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

For information on the MVS Message Service, refer to *MVS/ESA Planning: Operations* and the *MVS/ESA Application Development Guide: Assembler Language Programs*.

LANGTAB

```
►►──┬──────────────────────────────────────────────────────────────────────────►◄
    └LANGTAB(USS_table_name,language_code)──┘
```

Specifies the session-level USS table that VTAM uses for USS messages and commands. It also specifies the language that is passed to the application program. This operand provides the same function as the LANG operand, except it does not use the MVS Message Service.

uss_table_name
Specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

language_code
Specifies a 3-character code used to specify the language that is passed to the application program during logon processing. If a valid LANG operand is also specified and the MVS Message Service is active, this parameter is overridden by the language specified with the LANG operand. For a list of valid values and for details on how the application obtains this information, refer to *OS/390 eNetwork Communications Server: SNA Programming*.

Notes:

1. If you enter this operand without a USS table name, it removes the LANGTAB USS table (specified earlier) from the search hierarchy.
2. If you enter this operand without a language code, the application receives the language from the LANG operand of the MODEENT macroinstruction (specified by LOGMODE).

number

```
►►──┬─10─────┬──────────────────────────────────────────────────────────►◄
    └─number─┘
```

range: *1–255*

Specifies the number of times the test data should be returned to the terminal.

test_data

```
►►──┬──────────┬──────────────────────────────────────────────────────────►◄
    └─test_data─┘
```

Specifies the test data to be returned. Specify a character string of up to 247 characters, or the maximum message length of the terminal, whichever is smaller. If you do not specify a character string, VTAM sends out and returns a string consisting of the letters A–Z followed by the decimal numbers 0–9. (“ABCDEF...XYZ01...789”)

You can include blanks as part of this message if the string is surrounded by single quotes.

Appendix A. IBM-Supplied Tables

This appendix includes:
- The default logon mode table (ISTINCLM)
- The session-level USS table (ISTINCDT)
- The operation-level USS table (ISTINCNO)
- The APPN class-of-service (COS) definitions (COSAPPN)
- The APPN transmission group (TG) profile definitions (IBMTGPS)
- The message flooding table (ISTMSFLD)
- The CMIP services directory definition file (ACYDDF)

If you use this appendix to code replacements for or changes to any of these tables, be sure to refer to “Format of Macroinstructions” on page 10 for correct coding formats, especially when continuing lines of code.

Default Logon Mode Table (ISTINCLM)

```
         EJECT
ISTINCLM MODETAB
         EJECT
IBM3767  MODEENT LOGMODE=INTERACT,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'A0',COMPROT=X'3040',APPNCOS=#INTER
         EJECT
         TITLE 'TWXDEVPT'
*********************************************************************
*                                                                   *
*              TWX DEVICE WITH THE DCODE SET TO KEYBOARD            *
*              AND PRINTER.  THIS IS THE DEFAULT SETTING.           *
*                                                                   *
*********************************************************************
TWXDEVPT MODEENT LOGMODE=TWXDECPT,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'A0',COMPROT=X'3040',DCODE=X'00',          *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'TWXDEVDP'
*********************************************************************
*                                                                   *
*              TWX DEVICE WITH THE DCODE SET TO KEYBOARD            *
*              AND DISPLAY.                                         *
*                                                                   *
*********************************************************************
TWXDEVDP MODEENT LOGMODE=TWXDEVDP,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'A0',COMPROT=X'3040',DCODE=X'80',          *
               APPNCOS=#CONNECT
         EJECT
IBM3770  MODEENT LOGMODE=BATCH,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'A3',*
               SECPROT=X'A3',COMPROT=X'7080',APPNCOS=#BATCH
         EJECT
IBMS3270 MODEENT LOGMODE=S3270,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'71',*
               SECPROT=X'40',COMPROT=X'2000',                         *
               APPNCOS=#CONNECT
         EJECT
IBM3600  MODEENT LOGMODE=IBM3600,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'F1*
               ',SECPROT=X'F1',COMPROT=X'7000',APPNCOS=#CONNECT
         EJECT
IBM3650I MODEENT LOGMODE=INTRACT,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'B1*
               ',SECPROT=X'90',COMPROT=X'6000',APPNCOS=#INTER
         EJECT
IBM3650U MODEENT LOGMODE=INTRUSER,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'3*
               1',SECPROT=X'30',COMPROT=X'6000',APPNCOS=#INTER
         EJECT
```

© Copyright IBM Corp. 1984, 1999

Default Logon Mode Table

```
IBMS3650 MODEENT LOGMODE=IBMS3650,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'B*
               0',SECPROT=X'B0',COMPROT=X'4000',APPNCOS=#CONNECT
         EJECT
IBM3650P MODEENT LOGMODE=PIPELINE,FMPROF=X'04',TSPROF=X'04',PRIPROT=X'3*
               0',SECPROT=X'10',COMPROT=X'0000',APPNCOS=#CONNECT
         EJECT
IBM3660  MODEENT LOGMODE=SMAPPL,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'A0'*
               ,SECPROT=X'A0',COMPROT=X'0081',APPNCOS=#CONNECT
         EJECT
IBM3660A MODEENT LOGMODE=SMSNA100,FMPROF=X'00',TSPROF=X'00',PRIPROT=X'0*
               0',SECPROT=X'00',COMPROT=X'0000',APPNCOS=#CONNECT
         TITLE 'D6327801'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3278 MODEL 1 SCREEN                    *
*              PRIMARY SCREEN 12 X 40 (480)                         *
*              ALTERNATE SCREEN 12 X 80 (960)                       *
*                                                                   *
*********************************************************************
D6327801 MODEENT LOGMODE=D6327801,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'88F8',PSERVIC*
               =X'0200000000000C280C507F00',APPNCOS=#CONNECT
         TITLE 'D6327802'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3278 MODEL 2 SCREEN                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D6327802 MODEENT LOGMODE=D6327802,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'88F8',PSERVIC*
               =X'0200000000001850000007E00',APPNCOS=#CONNECT
         TITLE 'D6327803'
*********************************************************************
*                                                                   *
*              3276 SNA WITH      MODEL 3 SCREEN                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*                                                                   *
*********************************************************************
D6327803 MODEENT LOGMODE=D6327803,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'88F8',PSERVIC*
               =X'0200000000001850205 07F00',APPNCOS=#CONNECT
         TITLE 'D6327804'
*********************************************************************
*                                                                   *
*              3276 SNA WITH      MODEL 4 SCREEN                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 43 X 80 (3440)                      *
*                                                                   *
*********************************************************************
D6327804 MODEENT LOGMODE=D6327804,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'88F8',PSERVIC*
               =X'02000000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'D6327805'
*********************************************************************
*                                                                   *
*              3276 SNA WITH      MODEL 5 SCREEN                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 27 X 132 (3564)                     *
*                                                                   *
*********************************************************************
D6327805 MODEENT LOGMODE=D6327805,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'88F8',PSERVIC*
               =X'02000000000018501B847F00',APPNCOS=#CONNECT
         TITLE 'D6328904'
```

```
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3289 MODEL 4 PRINTER                   *
*                                                                   *
*********************************************************************
D6328904 MODEENT LOGMODE=D6328904,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'8787',PSERVIC*
               =X'03000000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'D6328902'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3289 MODEL 2 PRINTER                   *
*                                                                   *
*********************************************************************
D6328902 MODEENT LOGMODE=D6328902,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'8787',PSERVIC*
               =X'03000000000185018507F00',APPNCOS=#CONNECT
         TITLE 'D4A32781'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 1 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 12 X 40 (480)                         *
*              ALTERNATE SCREEN 12 X 80 (960)                       *
*                                                                   *
*********************************************************************
D4A32781 MODEENT LOGMODE=D4A32781,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'0200000000000C280C507F00',APPNCOS=#CONNECT
         TITLE 'D4A32782'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D4A32782 MODEENT LOGMODE=D4A32782,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000185000007E00',APPNCOS=#CONNECT
         TITLE 'LSK32782'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*              KATAKANA                                             *
*                                                                   *
*********************************************************************
LSK32782 MODEENT LOGMODE=LSK32782,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000185000007E00',LANG=X'11',               *
               APPNCOS=#CONNECT
         TITLE 'D4A32783'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 3 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*                                                                   *
*********************************************************************
D4A32783 MODEENT LOGMODE=D4A32783,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000185020507F00',APPNCOS=#CONNECT
         TITLE 'D4A32784'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 4 SCREEN (LOCAL SNA)        *
```

```
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 43 X 80 (3440)                      *
*                                                                   *
*********************************************************************
D4A32784 MODEENT LOGMODE=D4A32784,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'D4A32785'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 5 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 27 X 132 (3564)                     *
*                                                                   *
*********************************************************************
D4A32785 MODEENT LOGMODE=D4A32785,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000018501B847F00',APPNCOS=#CONNECT
         TITLE 'D4A32XX3'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)                            *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN TO BE DETERMINED BY APPLICATION     *
*                                                                   *
*********************************************************************
D4A32XX3 MODEENT LOGMODE=D4A32XX3,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000000000000300',APPNCOS=#CONNECT
         TITLE 'D4A32771'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH 3277 MODEL 1 SCREEN               *
*                                                                   *
*********************************************************************
D4A32771 MODEENT LOGMODE=D4A32771,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000000000000100',APPNCOS=#CONNECT
         TITLE 'D4A32772'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH 3277 MODEL 2 SCREEN               *
*                                                                   *
*********************************************************************
D4A32772 MODEENT LOGMODE=D4A32772,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02000000000000000000200',APPNCOS=#CONNECT
         TITLE 'D4C32781'
*********************************************************************
*                                                                   *
*              3274 MODEL 1C WITH MODEL 1 SCREEN(REMOTE SNA)        *
*              PRIMARY SCREEN 12 X 40 (480)                         *
*              ALTERNATE SCREEN 12 X 80 (960)                       *
*                                                                   *
*********************************************************************
D4C32781 MODEENT LOGMODE=D4C32781,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'0200000000000C280C507F00',APPNCOS=#CONNECT
         TITLE 'D4C32782'
*********************************************************************
*                                                                   *
*              3274 MODEL 1C WITH MODEL 2 SCREEN(REMOTE SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D4C32782 MODEENT LOGMODE=D4C32782,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
```

```
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02000000000185000007E00',APPNCOS=#CONNECT
         TITLE 'RSK32782'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C WITH MODEL 2 SCREEN(REMOTE SNA)      *
*              PRIMARY SCREEN 24 X 80 (1920)                      *
*              NO ALTERNATE SCREEN DEFINED                        *
*              KATAKANA                                           *
*                                                                 *
*******************************************************************
RSK32782 MODEENT LOGMODE=RSK32782,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02000000000185000007E00',LANG=X'11',               *
               APPNCOS=#CONNECT
         TITLE 'D4C32783'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C WITH MODEL 3 SCREEN(REMOTE SNA)      *
*              PRIMARY SCREEN 24 X 80 (1920)                      *
*              ALTERNATE SCREEN 32 X 80 (2560)                    *
*                                                                 *
*******************************************************************
D4C32783 MODEENT LOGMODE=D4C32783,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02000000000185020507F00',APPNCOS=#CONNECT
         TITLE 'D4C32784'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C WITH MODEL 4 SCREEN(REMOTE SNA)      *
*              PRIMARY SCREEN 24 X 80 (1920)                      *
*              ALTERNATE SCREEN 43 X 80 (3440)                    *
*                                                                 *
*******************************************************************
D4C32784 MODEENT LOGMODE=D4C32784,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02000000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'D4C32785'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C WITH MODEL 5 SCREEN(REMOTE SNA)      *
*              PRIMARY SCREEN 24 X 80 (1920)                      *
*              ALTERNATE SCREEN 27 X 132 (3564)                   *
*                                                                 *
*******************************************************************
D4C32785 MODEENT LOGMODE=D4C32785,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02000000000018501B847F00',APPNCOS=#CONNECT
         TITLE 'D4C32XX3'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C (REMOTE SNA)                         *
*              PRIMARY SCREEN 24 X 80 (1920)                      *
*              ALTERNATE SCREEN TO BE DETERMINED BY APPLICATION   *
*                                                                 *
*******************************************************************
D4C32XX3 MODEENT LOGMODE=D4C32XX3,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'02800000000000000000300',APPNCOS=#CONNECT
         TITLE 'D4C32771'
*******************************************************************
*                                                                 *
*              3274 MODEL 1C WITH 3277 MODEL 1 SCREEN             *
*                                                                 *
*******************************************************************
D4C32771 MODEENT LOGMODE=D4C32771,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
```

```
               =X'020000000000000000000100',APPNCOS=#CONNECT
         TITLE 'D4C32772'
*********************************************************************
*                                                                   *
*              3274 MODEL 1C WITH 3277 MODEL 2 SCREEN               *
*                                                                   *
*********************************************************************
D4C32772 MODEENT LOGMODE=D4C32772,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87F8',PSERVIC*
               =X'020000000000000000000200',APPNCOS=#CONNECT
         TITLE 'D4B32781'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 1 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 1 SCREEN                              *
*      3276 BSC WITH MODEL 1 SCREEN                                 *
*      PRIMARY SCREEN 12 X 40 (480)                                 *
*      ALTERNATE SCREEN 12 X 80 (960)                               *
*                                                                   *
*********************************************************************
D4B32781 MODEENT LOGMODE=D4B32781,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'0000000000000C280C507F00',APPNCOS=#CONNECT
         TITLE 'D4B32782'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 2 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 2 SCREEN                              *
*      3276 BSC WITH MODEL 2 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      NO ALTERNATE SCREEN DEFINED                                  *
*                                                                   *
*********************************************************************
D4B32782 MODEENT LOGMODE=D4B32782,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'000000000000185000007E00',APPNCOS=#CONNECT
         TITLE 'LNK32782'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 2 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 2 SCREEN                              *
*      3276 BSC WITH MODEL 2 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      NO ALTERNATE SCREEN DEFINED                                  *
*      KATAKANA                                                     *
*                                                                   *
*********************************************************************
LNK32782 MODEENT LOGMODE=LNK32782,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'000000000000185000007E00',LANG=X'11',               *
               APPNCOS=#CONNECT
         TITLE 'LNN32782'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 2 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 2 SCREEN                              *
*      3276 BSC WITH MODEL 2 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      NO ALTERNATE SCREEN DEFINED                                  *
*      NO LANGUAGE SPECIFIC FILTERING                               *
*                                                                   *
*********************************************************************
LNN32782 MODEENT LOGMODE=LNN32782,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'000000000000185000007E00',LANG=X'7F',               *
               APPNCOS=#CONNECT
         TITLE 'D4B32783'
```

```
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 3 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 3 SCREEN                              *
*      3276 BSC WITH MODEL 3 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      ALTERNATE SCREEN 32 X 80 (2560)                              *
*                                                                   *
*********************************************************************
D4B32783 MODEENT LOGMODE=D4B32783,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'00000000000185020507F00',APPNCOS=#CONNECT
         TITLE 'D4B32784'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 4 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 4 SCREEN                              *
*      3276 BSC WITH MODEL 4 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      ALTERNATE SCREEN 43 X 80 (3440)                              *
*                                                                   *
*********************************************************************
D4B32784 MODEENT LOGMODE=D4B32784,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'00000000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'D4B32785'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D WITH MODEL 5 SCREEN (LOCAL NON-SNA)         *
*      3274 1C BSC WITH MODEL 5 SCREEN                              *
*      3276 BSC WITH MODEL 5 SCREEN                                 *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      ALTERNATE SCREEN 27 X 132 (3564)                             *
*                                                                   *
*********************************************************************
D4B32785 MODEENT LOGMODE=D4B32785,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'00000000000018501B847F00',APPNCOS=#CONNECT
         TITLE 'D4B32XX3'
*********************************************************************
*                                                                   *
*      3274 MODEL 1B/1D (LOCAL NON-SNA)                             *
*      3274 1C BSC                                                  *
*      3276 BSC                                                     *
*      PRIMARY SCREEN 24 X 80 (1920)                                *
*      ALTERNATE SCREEN TO BE DETERMINED BY APPLICATION             *
*                                                                   *
*********************************************************************
D4B32XX3 MODEENT LOGMODE=D4B32XX3,FMPROF=X'02',TSPROF=X'02',PRIPROT=X'7*
               1',SECPROT=X'40',COMPROT=X'2000',RUSIZES=X'0000',PSERVIC*
               =X'008000000000000000000300',APPNCOS=#CONNECT
         TITLE 'SCS'
*********************************************************************
*                                                                   *
*      PRINTER WITH SNA CHARACTER SET                               *
*                                                                   *
*********************************************************************
SCS      MODEENT LOGMODE=SCS,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B1',  *
               SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C6',         *
               PSERVIC=X'01000000E100000000000000',                   *
               PSNDPAC=X'01',SRCVPAC=X'01',APPNCOS=#CONNECT
         TITLE 'DSC4K'
*********************************************************************
*                                                                   *
*      PRINTER WITH 4K BUFFER                                       *
*                                                                   *
*********************************************************************
```

Default Logon Mode Table

```
DSC4K    MODEENT LOGMODE=DSC4K,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B1',*
               SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'8787',          *
               PSERVIC=X'03000000000018502B507F00',                    *
               APPNCOS=#CONNECT
         TITLE 'DSC2K'
*********************************************************************
*                                                                   *
*      PRINTER WITH 2K BUFFER                                       *
*                                                                   *
*********************************************************************
DSC2K    MODEENT LOGMODE=DSC2K,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B1',*
               SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'8787',          *
               PSERVIC=X'03000000000185018507F00',                     *
               APPNCOS=#CONNECT
         TITLE 'BAT13790'
*********************************************************************
*                                                                   *
*      3790 BATCH                                                   *
*                                                                   *
*********************************************************************
BAT13790 MODEENT LOGMODE=BAT13790,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'00',SECPROT=X'00',COMPROT=X'0000',            *
               RUSIZES=X'0000',APPNCOS=#BATCH
         TITLE 'EMU3790'
*********************************************************************
*                                                                   *
*      3790 IN DATA STREAM COMPATIBILITY MODE                       *
*                                                                   *
*********************************************************************
EMU3790  MODEENT LOGMODE=EMU3790,FMPROF=X'03',TSPROF=X'03',            *
               PRIPROT=X'B1',SECPROT=X'B0',COMPROT=X'3080',            *
               RUSIZES=X'85C7',PSERVIC=X'020000000000000000000200',    *
               APPNCOS=#CONNECT
         TITLE 'RJE3790A'
*********************************************************************
*                                                                   *
*      3790 RJE                                                     *
*                                                                   *
*********************************************************************
RJE3790A MODEENT LOGMODE=RJE3790A,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'A3',SECPROT=X'A1',COMPROT=X'7080',            *
               RUSIZES=X'8585',PSERVIC=X'01106000F100800000010040',    *
               APPNCOS=#CONNECT
         TITLE 'RJE3790B'
*********************************************************************
*                                                                   *
*      3790 RJE                                                     *
*                                                                   *
*********************************************************************
RJE3790B MODEENT LOGMODE=RJE3790B,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'A3',SECPROT=X'A1',COMPROT=X'7080',            *
               RUSIZES=X'8585',PSERVIC=X'01102000F100800000010040',    *
               APPNCOS=#CONNECT
         TITLE 'BAT23790'
*********************************************************************
*                                                                   *
*      3790 BATCH                                                   *
*                                                                   *
*********************************************************************
BAT23790 MODEENT LOGMODE=BAT23790,FMPROF=X'03',TSPROF=X'04',           *
               PRIPROT=X'B1',SECPROT=X'B0',COMPROT=X'7080',            *
               RUSIZES=X'8585',PSERVIC=X'013100000000000000000000',    *
               APPNCOS=#BATCH
         TITLE 'BLK3790'
*********************************************************************
*                                                                   *
*      3790 BULK PRINT                                              *
```

```
*                                                                    *
**********************************************************************
BLK3790  MODEENT LOGMODE=BLK3790,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'B0',COMPROT=X'3080',          *
               RUSIZES=X'8585',PSERVIC=X'030000000000000000000000',  *
               APPNCOS=#CONNECT
         TITLE 'SCS3790'
**********************************************************************
*                                                                    *
*      3790 WITH SNA CHARACTER SET                                   *
*                                                                    *
**********************************************************************
SCS3790  MODEENT LOGMODE=SCS3790,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'B0',COMPROT=X'3080',          *
               RUSIZES=X'8585',PSERVIC=X'010000000000000000000000',  *
               APPNCOS=#CONNECT
         TITLE 'EMUDPCX'
**********************************************************************
*                                                                    *
*      3790 IN DPCX EMULATION MODE                                   *
*                                                                    *
**********************************************************************
EMUDPCX  MODEENT LOGMODE=EMUDPCX,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'B0',COMPROT=X'3080',          *
               RUSIZES=X'85C7',PSERVIC=X'020000000000000000000200',  *
               APPNCOS=#CONNECT
         TITLE 'DSILGMOD'
**********************************************************************
*                                                                    *
* DSILGMOD  LOGMODE TABLE FOR BSC,LOCAL,SDLC 3275,3277,3278,3279     *
*           MODEL 2 OR 12, 24 X 80 SCREEN. MAY BE USED TO RUN        *
*              MODELS 3, 4, 5, 2C OR 3C AS MODEL 2                   *
*           ALSO FOR 3284, 3286, 3287, 3288, 3289 PRINTERS           *
*           THROUGH A 3271, 3272, 3274, 3275, OR 3276 CONTROLLER     *
*                                                                    *
**********************************************************************
DSILGMOD MODEENT LOGMODE=DSILGMOD,FMPROF=X'02',TSPROF=X'02',         *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',          *
               RUSIZES=X'0000',PSERVIC=X'000000000000000000000200',  *
               APPNCOS=#CONNECT
         TITLE 'ISTNLDM'
**********************************************************************
*        NLDM LOGMODE FOR LU - LU SESSION WITH NCCF                  *
**********************************************************************
ISTNLDM  MODEENT LOGMODE=ISTNLDM,FMPROF=X'02',TSPROF=X'03',          *
               PRIPROT=X'30',SECPROT=X'40',COMPROT=X'0000',          *
               SSNDPAC=X'02',RUSIZES=X'0000',                        *
               PSERVIC=X'000000000000000000000000',                  *
               APPNCOS=#CONNECT
         TITLE 'D329001'
**********************************************************************
*        LOGMODE TABLE ENTRY FOR THE 3290 TERMINAL                   *
*              OR EXTENDED DATA STREAM TERMINAL OFF 3274-1A          *
*              PRIMARY SCREEN SIZE 24 X 80                           *
*              ALTERNATE SCREEN SIZE 62 X 160                        *
**********************************************************************
D329001  MODEENT LOGMODE=D329001,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               RUSIZES=X'8787',                                      *
               PSERVIC=X'02800000000018503EA07F00',                  *
               APPNCOS=#CONNECT
         TITLE 'NSX32702'
**********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH           *
*           EXTENDED DATA STREAMS (3278 OR 3279).                    *
*           SCREEN SIZE IS 24 X 80.                                  *
**********************************************************************
```

```
NSX32702 MODEENT LOGMODE=NSX32702,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000185000007E00',                    *
               APPNCOS=#CONNECT
         TITLE 'NSX32703'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH          *
*           EXTENDED DATA STREAMS (3278 OR 3279).                   *
*           PRIMARY SCREEN 24 X 80                                  *
*           ALTERNATE SCREEN 32 X 80                                *
*********************************************************************
NSX32703 MODEENT LOGMODE=NSX32703,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000185020507F00',                    *
               APPNCOS=#CONNECT
         TITLE 'NSX32704'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH          *
*           EXTENDED DATA STREAMS (3278 OR 3279).                   *
*           PRIMARY SCREEN 24 X 80                                  *
*           ALTERNATE SCREEN 43 X 80                                *
*********************************************************************
NSX32704 MODEENT LOGMODE=NSX32704,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000018502B507F00',                   *
              APPNCOS=#CONNECT
         TITLE 'NSX32705'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH          *
*           EXTENDED DATA STREAMS (3278 OR 3279).                   *
*           PRIMARY SCREEN 24 X 80                                  *
*           ALTERNATE SCREEN 27 X 132                               *
*********************************************************************
NSX32705 MODEENT LOGMODE=NSX32705,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000018501B847F00',                   *
               APPNCOS=#CONNECT
         TITLE 'SNX32702'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (3278 OR 3279).              *
*           SCREEN SIZE IS 24 X 80.                                 *
*********************************************************************
SNX32702 MODEENT LOGMODE=SNX32702,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',                                       *
               PSERVIC=X'02800000000185000007E00',                    *
               APPNCOS=#CONNECT
         TITLE 'SNX32703'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (MOD3).                      *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*********************************************************************
SNX32703 MODEENT LOGMODE=SNX32703,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',                                       *
               PSERVIC=X'02800000000185020507F00',                    *
               APPNCOS=#CONNECT
         TITLE 'SNX32704'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
```

```
*           WITH EXTENDED DATA STREAMS (MOD4).                      *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 43 X 80 (3440)                      *
*********************************************************************
SNX32704 MODEENT LOGMODE=SNX32704,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',                                       *
               PSERVIC=X'02800000000018502B507F00',                   *
               APPNCOS=#CONNECT
         TITLE 'SNX32705'
*********************************************************************
*                                                                   *
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (MOD5).                      *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 27 X 132 (3564)                     *
*                                                                   *
*********************************************************************
SNX32705 MODEENT LOGMODE=SNX32705,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',                                       *
               PSERVIC=X'02800000000018501B847F00',                   *
               APPNCOS=#CONNECT
         TITLE 'LSX32702'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              WITH EXTENDED DATA STREAMS (MOD2)                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
LSX32702 MODEENT LOGMODE=LSX32702,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018500000 7E00',APPNCOS=#CONNECT
         TITLE 'LSX32703'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 3 SCREEN (LOCAL SNA)        *
*              WITH EXTENDED DATA STREAMS (MOD3)                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*                                                                   *
*********************************************************************
LSX32703 MODEENT LOGMODE=LSX32703,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018502050 7F00',APPNCOS=#CONNECT
         TITLE 'LSX32704'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 4 SCREEN (LOCAL SNA)        *
*              WITH EXTENDED DATA STREAMS (MOD4)                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 43 X 80 (3440)                      *
*                                                                   *
*********************************************************************
LSX32704 MODEENT LOGMODE=LSX32704,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018502B507F00',APPNCOS=#CONNECT
         TITLE 'LSX32705'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A WITH MODEL 5 SCREEN (LOCAL SNA)        *
*              WITH EXTENDED DATA STREAMS (MOD5)                    *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 27 X 132 (3564)                     *
*                                                                   *
```

Default Logon Mode Table

```
*********************************************************************
LSX32705 MODEENT LOGMODE=LSX32705,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018501B847F00',APPNCOS=#CONNECT
         TITLE 'NED32702'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH          *
*           EXTENDED DATA STREAMS (3278 OR 3279).                   *
*           SCREEN SIZE IS 24 X 80.                                 *
*           LANGUAGE IS ENGLISH.                                    *
*           QUERY FOR DOUBLE BYTE CAPABILITY.                       *
*********************************************************************
NED32702 MODEENT LOGMODE=NED32702,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000018500000 7E00',LANG=X'81',       *
               APPNCOS=#CONNECT
         TITLE 'NKD32702'
*********************************************************************
* NAME:  NON-SNA KATAKANA, DOUBLE BYTE CAPABLE, 3270-2              *
*        LOGMODE TABLE ENTRY FOR NON-SNA 3270 DEVICES WITH          *
*           EXTENDED DATA STREAMS (3278 OR 3279).                   *
*           SCREEN SIZE IS 24 X 80.                                 *
*           LANGUAGE IS KATAKANA.                                   *
*           QUERY FOR DOUBLE BYTE CAPABILITY.                       *
*********************************************************************
NKD32702 MODEENT LOGMODE=NKD32702,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',                                       *
               PSERVIC=X'00800000000018500000 7E00',LANG=X'91',       *
               APPNCOS=#CONNECT
         TITLE 'LED32702'
*********************************************************************
*                                                                   *
*        LOGMODE TABLE ENTRY FOR LOCAL SNA                          *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*              EXTENDED DATA STREAMS                                *
*              ENGLISH LANGUAGE                                     *
*              QUERY FOR DOUBLE BYTE CAPABILITY                     *
*********************************************************************
LED32702 MODEENT LOGMODE=LED32702,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018500000 7E00',LANG=X'81',              *
               APPNCOS=#CONNECT
         TITLE 'LKD32702'
*********************************************************************
*                                                                   *
*        LOGMODE TABLE ENTRY FOR LOCAL SNA                          *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*              EXTENDED DATA STREAMS                                *
*              KATAKANA LANGUAGE                                    *
*              QUERY FOR DOUBLE BYTE CAPABILITY                     *
*********************************************************************
LKD32702 MODEENT LOGMODE=LKD32702,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'02800000000018500000 7E00',LANG=X'91',              *
               APPNCOS=#CONNECT
         TITLE 'LND32702'
*********************************************************************
*                                                                   *
*        LOGMODE TABLE ENTRY FOR LOCAL SNA                          *
*              3274 MODEL 1A WITH MODEL 2 SCREEN (LOCAL SNA)        *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
```

```
*              NO ALTERNATE SCREEN DEFINED                          *
*              EXTENDED DATA STREAMS                                *
*              NO LANGUAGE SPECIFIC FILTERING                       *
*              QUERY FOR DOUBLE BYTE CAPABILITY                     *
*********************************************************************
LND32702 MODEENT LOGMODE=LND32702,FMPROF=X'03',TSPROF=X'03',PRIPROT=X'B*
               1',SECPROT=X'90',COMPROT=X'3080',RUSIZES=X'87C7',PSERVIC*
               =X'028000000000185000007E00',LANG=X'FF',               *
               APPNCOS=#CONNECT
         TITLE 'SED32702'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (3278 OR 3279).              *
*           SCREEN SIZE IS 24 X 80.                                 *
*           LANGUAGE IS ENGLISH.                                    *
*           QUERY FOR DOUBLE BYTE CAPABILITY                        *
*********************************************************************
SED32702 MODEENT LOGMODE=SED32702,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',                                        *
               PSERVIC=X'028000000000185000007E00',LANG=X'81',         *
               APPNCOS=#CONNECT
         TITLE 'SKD32702'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (3278 OR 3279).              *
*           SCREEN SIZE IS 24 X 80.                                 *
*           LANGUAGE IS KATAKANA.                                   *
*           QUERY FOR DOUBLE BYTE CAPABILITY                        *
*********************************************************************
SKD32702 MODEENT LOGMODE=SKD32702,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',                                        *
               PSERVIC=X'028000000000185000007E00',LANG=X'91',         *
               APPNCOS=#CONNECT
         TITLE 'SND32702'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR REMOTE SNA 3270 DEVICES            *
*           WITH EXTENDED DATA STREAMS (3278 OR 3279).              *
*           SCREEN SIZE IS 24 X 80.                                 *
*           NO LANGUAGE SPECIFIC FILTERING                          *
*           QUERY FOR DOUBLE BYTE CAPABILITY                        *
*********************************************************************
SND32702 MODEENT LOGMODE=SND32702,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',                                        *
               PSERVIC=X'028000000000185000007E00',LANG=X'FF',         *
               APPNCOS=#CONNECT
         TITLE 'SNASVCMG'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR RESOURCES CAPABLE OF ACTING        *
*                                AS LU 6.2 DEVICES
*********************************************************************
SNASVCMG MODEENT LOGMODE=SNASVCMG,TYPE=0,FMPROF=X'13',TSPROF=X'07',    *
               PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'D0B1',            *
               RUSIZES=X'9797',ENCR=B'0000',                           *
               PSERVIC=X'060200000000000000002300',                    *
               APPNCOS=SNASVCMG                                      *
                              SPECIFY SYNC LEVEL = CONFIRM
         TITLE '#BATCH'
*********************************************************************
*                                                                   *
*        LOGMODE TABLE FOR BATCH SESSIONS ON RESOURCES CAPABLE      *
*        OF ACTING AS LU 6.2 DEVICES                                *
*                                                                   *
*********************************************************************
#BATCH   MODEENT LOGMODE=#BATCH,FMPROF=X'13',TSPROF=X'07',            *
```

```
              ENCR=B'0000',SSNDPAC=3,RUSIZES=X'F7F7',                *
              SRCVPAC=3,PSNDPAC=3,APPNCOS=#BATCH
        TITLE '#INTER'
*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR BATCH COMPRESSION SESSIONS ON RESOURCES   *
*       CAPABLE OF ACTING AS LU 6.2 DEVICES                         *
*                                                                   *
*********************************************************************
#BATCHC   MODEENT LOGMODE=#BATCHC,FMPROF=X'13',TSPROF=X'07',          *
              ENCR=B'0000',SSNDPAC=3,RUSIZES=X'F7F7',                *
              SRCVPAC=3,PSNDPAC=3,APPNCOS=#BATCH
        TITLE '#INTERC'

*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR INTERACTIVE SESSIONS ON RESOURCES         *
*       CAPABLE OF ACTING AS LU 6.2 DEVICES                         *
*                                                                   *
*********************************************************************
#INTER   MODEENT LOGMODE=#INTER,FMPROF=X'13',TSPROF=X'07',            *
              ENCR=B'0000',SSNDPAC=7,RUSIZES=X'F7F7',                *
              SRCVPAC=7,PSNDPAC=7,APPNCOS=#INTER
        TITLE '#BATCHSC'
*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR INTERACTIVE COMPRESSION SESSIONS ON         *
*       RESOURCES CAPABLE OF ACTING AS LU 6.2 DEVICES               *
*                                                                   *
*********************************************************************
#INTERC   MODEENT LOGMODE=#INTERC,FMPROF=X'13',TSPROF=X'07',          *
              ENCR=B'0000',SSNDPAC=7,RUSIZES=X'F7F7',                *
              SRCVPAC=7,PSNDPAC=7,APPNCOS=#INTER

*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR BATCH SESSIONS REQUIRING SECURE           *
*       TRANSPORT ON RESOURCES CAPABLE OF ACTING AS LU 6.2          *
*       DEVICES                                                     *
*                                                                   *
*********************************************************************
#BATCHSC MODEENT LOGMODE=#BATCHSC,FMPROF=X'13',TSPROF=X'07',         *
              ENCR=B'0000',SSNDPAC=3,RUSIZES=X'F7F7',                *
              SRCVPAC=3,PSNDPAC=3,APPNCOS=#BATCHSC
        TITLE '#INTERSC'
*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR INTERACTIVE SESSIONS REQUIRING            *
*       SECURE TRANSPORT ON RESOURCES CAPABLE OF ACTING AS          *
*       LU 6.2 DEVICES                                              *
*                                                                   *
*********************************************************************
#INTERSC MODEENT LOGMODE=#INTERSC,FMPROF=X'13',TSPROF=X'07',         *
              ENCR=B'0000',SSNDPAC=7,RUSIZES=X'F7F7',                *
              SRCVPAC=7,PSNDPAC=7,APPNCOS=#INTERSC
        TITLE 'CPSVCMG'
*********************************************************************
*                                                                   *
*       LOGMODE TABLE FOR CP-CP SESSIONS ON RESOURCES CAPABLE       *
*       OF ACTING AS LU 6.2 DEVICES                                 *
*                                                                   *
*********************************************************************
CPSVCMG  MODEENT LOGMODE=CPSVCMG,                                    *
              RUSIZES=X'9797',ENCR=B'0000',                          *
              SSNDPAC=7,SRCVPAC=7,PSNDPAC=7,                         *
              APPNCOS=CPSVCMG
*********************************************************************
```

```
*                                                                     *
*        LOGMODE TABLE ENTRY THAT SUPPLIES A DEFAULT COS              *
*        AND USES LU 6.2 DEVICE CHARACTERISTICS                       *
*                                                                     *
***********************************************************************
ISTCOSDF MODEENT LOGMODE=ISTCOSDF,FMPROF=X'13',                        *
               TSPROF=X'07',PRIPROT=X'B0',SECPROT=X'B0',               *
               COMPROT=X'D0B1',PSERVIC=X'06020000000000000000300',     *
               RUSIZES=X'8989',ENCR=B'0000',TYPE=0,                    *
               APPNCOS=#CONNECT
***********************************************************************
*                                                                     *
*        LOGMODE TABLE ENTRY FOR AS/400 PC SUPPORT                    *
*        CAPABLE OF ACTING AS LU 6.2 DEVICES                          *
*                                                                     *
***********************************************************************
QPCSUPP  MODEENT LOGMODE=QPCSUPP,FMPROF=X'13',TSPROF=X'07',            *
               ENCR=B'0000',SSNDPAC=7,                                 *
               SRCVPAC=7,PSNDPAC=7,                                    *
               APPNCOS=#CONNECT
***********************************************************************
*                                                                     *
*        LOGMODE TABLE ENTRY FOR DRDA SUPPORT                         *
*                                                                     *
***********************************************************************
IBMRDB   MODEENT LOGMODE=IBMRDB,   DRDA DEFAULT MODE                   *
               TYPE=0,             NEGOTIABLE BIND                     *
               PSNDPAC=X'00',      PRIMARY SEND PACING COUNT           *
               SSNDPAC=X'02',      SECONDARY SEND PACING COUNT         *
               SRCVPAC=X'00',      SECONDARY RECEIVE PACING COUNT      *
               RUSIZES=X'8989',    RUSIZES IN-4K    OUT-4K             *
               FMPROF=X'13',       LU6.2 FM PROFILE                    *
               TSPROF=X'07',       LU6.2 TS PROFILE                    *
               PRIPROT=X'B0',      LU6.2 PRIMARY PROTOCOLS             *
               SECPROT=X'B0',      LU6.2 SECONDARY PROTOCOLS           *
               COMPROT=X'50A5',    LU6.2 COMMON PROTOCOLS              *
               PSERVIC=X'06020000000000000102F00',   LU6.2 LU TYPE     *
               APPNCOS=#CONNECT
         TITLE 'CPSVRMGR'
***********************************************************************
*                                                                     *
*        LOGMODE TABLE FOR SESSIONS BETWEEN A DLS (DEPENDENT LU       *
*        SERVER) AND A DLR (DEPENDENT LU REQUESTOR).                  *
*                                                                     *
***********************************************************************
CPSVRMGR MODEENT LOGMODE=CPSVRMGR,                                     *
               ENCR=B'0000',                                           *
               RUSIZES=X'9797',                                        *
               SSNDPAC=7,SRCVPAC=7,PSNDPAC=7,                          *
               APPNCOS=SNASVCMG
         TITLE 'SNACKETS'
***********************************************************************
*                                                                     *
*        LOGMODE ENTRY FOR MPTN ANYNET SOCKETS OVER SNA               *
*                                                                     *
***********************************************************************
SNACKETS MODEENT LOGMODE=SNACKETS,FMPROF=X'13',TSPROF=X'07',           *
               ENCR=B'0000',SSNDPAC=7,RUSIZES=X'F8F8',                 *
               SRCVPAC=7,PSNDPAC=7,APPNCOS=#INTER
*
         TITLE 'DLUR'
***********************************************************************
*                                                                     *
*        LOGMODE ENTRY FOR 3174 DLUR                                  *
*                                                                     *
***********************************************************************
DLUR     MODEENT LOGMODE=DLUR,FMPROF=X'03',TSPROF=X'03',
```

```
                PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',
                SSNDPAC=X'04',PSNDPAC=X'04',RUSIZES=X'8787',
                PSERVIC=X'02800000000185000007E00',
                APPNCOS=#CONNECT
*
          TITLE 'SNADAUTH'
*********************************************************************
*                                                                   *
*         LOGMODE ENTRY FOR DCE
*                                                                   *
*********************************************************************
SNADAUTH MODEENT LOGMODE=SNADAUTH, PSNDPAC=X'03',
                SSNDPAC=X'03', SRCVPAC=X'03', ENCR=B'0000'
          MODEEND ,                    END OF DEFAULT TABLE ENTRIES
          END                        , END OF ISTINCLM
```

Session-Level USS Table (ISTINCDT)

```
          EJECT
ISTINCDT USSTAB   TABLE=STDTRANS,FORMAT=DYNAMIC
          SPACE 4
LOGON     USSCMD   CMD=LOGON,FORMAT=PL1
          USSPARM  PARM=APPLID
          USSPARM  PARM=LOGMODE
          USSPARM  PARM=DATA
          EJECT
LOGOFF    USSCMD   CMD=LOGOFF,FORMAT=PL1
          USSPARM  PARM=APPLID
          USSPARM  PARM=TYPE,DEFAULT=UNCOND
          USSPARM  PARM=HOLD,DEFAULT=YES
          EJECT
IBMTEST   USSCMD   CMD=IBMTEST,FORMAT=BAL
          USSPARM  PARM=P1,DEFAULT=10
          USSPARM  PARM=P2,DEFAULT=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
          EJECT
USSMSG01 USSMSG   MSG=1,TEXT='INVALID % COMMAND SYNTAX'
USSMSG02 USSMSG   MSG=2,TEXT='% COMMAND UNRECOGNIZED'
USSMSG03 USSMSG   MSG=3,TEXT='% PARAMETER EXTRANEOUS'
USSMSG04 USSMSG   MSG=4,TEXT='% PARAMETER VALUE %(2) NOT VALID'
USSMSG05 USSMSG   MSG=5,TEXT='UNSUPPORTED FUNCTION'
USSMSG06 USSMSG   MSG=6,TEXT='SEQUENCE ERROR'
USSMSG07 USSMSG   MSG=7,TEXT='%(1) UNABLE TO ESTABLISH SESSION - %(2) F-
                AILED WITH SENSE %(3)'
USSMSG08 USSMSG   MSG=8,TEXT='INSUFFICIENT STORAGE'
USSMSG09 USSMSG   MSG=9,TEXT='MAGNETIC CARD DATA ERROR'
USSMSG11 USSMSG   MSG=11,TEXT='% SESSIONS ENDED'
USSMSG12 USSMSG   MSG=12,TEXT='REQUIRED PARAMETER OMITTED'
USSMSG13 USSMSG   MSG=13,TEXT='IBMECHO % '
USSMSG14 USSMSG   MSG=14,TEXT='USS MESSAGE % NOT DEFINED'
          EJECT
STDTRANS DC       X'000102030440060708090A0B0C0D0E0F'
          DC       X'101112131415161718191A1B1C1D1E1F'
          DC       X'202122232425262728292A2B2C2D2E2F'
          DC       X'303132333435363738393A3B3C3D3E3F'
          DC       X'404142434445464748494A4B4C4D4E4F'
          DC       X'505152535455565758595A5B5C5D5E5F'
          DC       X'606162636465666768696A6B6C6D6E6F'
          DC       X'707172737475767778797A7B7C7D7E7F'
          DC       X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
          DC       X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
          DC       X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'
          DC       X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'
          DC       X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
          DC       X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
          DC       X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
```

```
         DC       X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
END      USSEND
         END     ,             END OF ASSEMBLY
```

Operation-Level USS Table (ISTINCNO)

```
*************************************************************
* DISPLAY COMMAND(NETWORK OPERATOR SERVICES)
*************************************************************
   USSCMD  CMD=DISPLAY,FORMAT=BAL
   USSPARM PARM=A,REP=SCOPE,VALUE=ACT
   USSPARM PARM=ACT,REP=SCOPE,VALUE=ACT
   USSPARM PARM=ACTONLY,REP=SCOPE,VALUE=ACTONLY
   USSPARM PARM=ADJCLUST,REP=VERB,VALUE=ADJCLUST
   USSPARM PARM=ADJCP,REP=VERB,VALUE=ADJCP
   USSPARM PARM=ADJSSCPS,REP=VERB,VALUE=ADJSSCPS
   USSPARM PARM=APING,REP=VERB,VALUE=APING
   USSPARM PARM=APINGDTP,REP=VERB,VALUE=APINGDTP
   USSPARM PARM=APPLS,REP=VERB,VALUE=APPLS
   USSPARM PARM=APPNTOSA,REP=VERB,VALUE=APPNTOSA
   USSPARM PARM=BFRUSE,REP=VERB,VALUE=BFRUSE
   USSPARM PARM=BNCOSMAP,REP=VERB,VALUE=BNCOSMAP
   USSPARM PARM=CDRMS,REP=VERB,VALUE=CDRMS
   USSPARM PARM=CDRSCS,REP=VERB,VALUE=CDRSCS
   USSPARM PARM=CLSTRS,REP=VERB,VALUE=CLSTRS
   USSPARM PARM=CNOS,REP=VERB,VALUE=DCNOS
   USSPARM PARM=CONCT,REP=SCOPE,VALUE=CONCT
   USSPARM PARM=CONVID,REP=VERB,VALUE=DCONVID
   USSPARM PARM=COS,REP=VERB,VALUE=COS
   USSPARM PARM=CPS,REP=VERB,VALUE=CPS
   USSPARM PARM=CPSTAB,REP=VERB,VALUE=CPS
   USSPARM PARM=DIRECTRY,REP=VERB,VALUE=DIRECTRY
   USSPARM PARM=DISK,REP=VERB,VALUE=DISK
   USSPARM PARM=DLUR,REP=VERB,VALUE=DLURS
   USSPARM PARM=DLURS,REP=VERB,VALUE=DLURS
   USSPARM PARM=E,REP=SCOPE,VALUE=ALL
   USSPARM PARM=EVERY,REP=SCOPE,VALUE=ALL
   USSPARM PARM=EXIT,REP=VERB,VALUE=EXIT
   USSPARM PARM=GROUPS,REP=VERB,VALUE=GROUPS
   USSPARM PARM=I,REP=SCOPE,VALUE=INACT
   USSPARM PARM=ID
   USSPARM PARM=INACT,REP=SCOPE,VALUE=INACT
   USSPARM PARM=INACTONL,REP=SCOPE,VALUE=INACTONLY
   USSPARM PARM=LINE,REP=VERB,VALUE=LINES
   USSPARM PARM=LINES,REP=VERB,VALUE=LINES
   USSPARM PARM=LMTBL,REP=VERB,VALUE=DLMTBL
   USSPARM PARM=LUGROUPS,REP=VERB,VALUE=LUGROUPS
   USSPARM PARM=MAJNODES,REP=VERB,VALUE=MAJNODES
   USSPARM PARM=MODELS,REP=VERB,VALUE=MODELS
   USSPARM PARM=N,REP=SCOPE,VALUE=ONLY
   USSPARM PARM=NCPSTOR,REP=VERB,VALUE=NCPSTOR
   USSPARM PARM=NETID
   USSPARM PARM=NETSRVR,REP=VERB,VALUE=NETSRVR
   USSPARM PARM=NONE,REP=SCOPE,VALUE=ONLY
   USSPARM PARM=P,REP=VERB,VALUE=PATHS
   USSPARM PARM=PATHS,REP=VERB,VALUE=PATHS
   USSPARM PARM=PATHTAB,REP=VERB,VALUE=PATHTAB
   USSPARM PARM=PEND,REP=SCOPE,VALUE=PENDING
   USSPARM PARM=PENDING,REP=VERB,VALUE=PENDING
   USSPARM PARM=RESET,REP=SCOPE,VALUE=RESET
   USSPARM PARM=ROUTE,REP=VERB,VALUE=ROUTE
   USSPARM PARM=RSCLIST,REP=VERB,VALUE=RSCLIST
   USSPARM PARM=SATOAPPN,REP=VERB,VALUE=SATOAPPN
   USSPARM PARM=SCOPE
   USSPARM PARM=SESSION,REP=VERB,VALUE=SESSIONS
   USSPARM PARM=SESSIONS,REP=VERB,VALUE=SESSIONS
   USSPARM PARM=STATIONS,REP=VERB,VALUE=STATIONS
```

```
   USSPARM PARM=STATS,REP=VERB,VALUE=STATS
   USSPARM PARM=STORUSE,REP=VERB,VALUE=STORUSE
   USSPARM PARM=TABLE,REP=VERB,VALUE=DTABLE
   USSPARM PARM=TERMS,REP=VERB,VALUE=TERMS
   USSPARM PARM=TGP,REP=VERB,VALUE=TGPS
   USSPARM PARM=TGPS,REP=VERB,VALUE=TGPS
   USSPARM PARM=TOPO,REP=VERB,VALUE=TOPO
   USSPARM PARM=TRACE,REP=VERB,VALUE=TRACES
   USSPARM PARM=TRACES,REP=VERB,VALUE=TRACES
   USSPARM PARM=TRL,REP=VERB,VALUE=TRL
   USSPARM PARM=TSOUSER,REP=VERB,VALUE=TSOUSER
   USSPARM PARM=U,REP=VERB,VALUE=TSOUSER
   USSPARM PARM=USERVAR,REP=VERB,VALUE=USERVAR
   USSPARM PARM=VTAMOPTS,REP=VERB,VALUE=VTAMOPTS
*************************************************************
* DISPLAY ADJCP COMMAND
*************************************************************
   USSCMD  CMD=ADJCP,FORMAT=BAL
*************************************************************
* DISPLAY ADJSSCPS COMMAND
*************************************************************
   USSCMD  CMD=ADJSSCPS,FORMAT=BAL
   USSPARM PARM=CDRM
   USSPARM PARM=CDRSC
   USSPARM PARM=SCOPE,DEFAULT=ONLY
   USSPARM PARM=MAX
*************************************************************
* DISPLAY APPLS COMMAND
*************************************************************
   USSCMD  CMD=APPLS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY APING COMMAND
*************************************************************
   USSCMD  CMD=APING,FORMAT=BAL
   USSPARM PARM=CONSEC,DEFAULT=1
   USSPARM PARM=ECHO,DEFAULT=YES
   USSPARM PARM=ITER,DEFAULT=2
   USSPARM PARM=LOGMODE,DEFAULT=#INTER
   USSPARM PARM=PASSWORD
   USSPARM PARM=SIZE,DEFAULT=100
   USSPARM PARM=TP,DEFAULT=APINGD
   USSPARM PARM=USERID
*************************************************************
* DISPLAY APINGDTP COMMAND
*************************************************************
   USSCMD  CMD=APINGDTP,FORMAT=BAL
*************************************************************
* DISPLAY APPNTOSA COMMAND
*************************************************************
   USSCMD  CMD=APPNTOSA,FORMAT=BAL
*************************************************************
* DISPLAY BFRUSE COMMAND
*************************************************************
   USSCMD  CMD=BFRUSE,FORMAT=BAL
   USSPARM PARM=BUF,REP=BUFFER
   USSPARM PARM=BUFFER,DEFAULT=*
*************************************************************
* DISPLAY BNCOSMAP COMMAND
*************************************************************
   USSCMD  CMD=BNCOSMAP,FORMAT=BAL
*************************************************************
* DISPLAY CDRMS COMMAND
*************************************************************
   USSCMD  CMD=CDRMS,FORMAT=BAL
   USSPARM PARM=MAX
```

```
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY CDRSCS COMMAND
*************************************************************
   USSCMD  CMD=CDRSCS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY CLSTRS COMMAND
*************************************************************
   USSCMD  CMD=CLSTRS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=OWNER
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY CNOS COMMAND
*************************************************************
   USSCMD  CMD=DCNOS,FORMAT=BAL
   USSPARM PARM=LOG,REP=LOGMODE
   USSPARM PARM=LOGMODE
   USSPARM PARM=LU,REP=LUNAME
   USSPARM PARM=LUNAME
*************************************************************
* DISPLAY CONVID COMMAND
*************************************************************
   USSCMD  CMD=DCONVID,FORMAT=BAL
   USSPARM PARM=ETIME,DEFAULT=0
   USSPARM PARM=LOG,REP=LOGMODE
   USSPARM PARM=LOGMODE
   USSPARM PARM=LU,REP=LUNAME
   USSPARM PARM=LUNAME
*************************************************************
* DISPLAY COS COMMAND
*************************************************************
   USSCMD  CMD=COS,FORMAT=BAL
*************************************************************
* DISPLAY CPS COMMAND
*************************************************************
   USSCMD CMD=CPS,FORMAT=BAL
*************************************************************
* DISPLAY DIRECTRY COMMAND
*************************************************************
   USSCMD  CMD=DIRECTRY,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ONLY
*************************************************************
* DISPLAY DISK COMMAND
*************************************************************
   USSCMD  CMD=DISK,FORMAT=BAL
*************************************************************
* DISPLAY EXIT COMMAND
*************************************************************
   USSCMD  CMD=EXIT,FORMAT=BAL
   USSPARM PARM=MAX
*************************************************************
* DISPLAY GROUPS COMMAND
*************************************************************
   USSCMD  CMD=GROUPS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY LINES COMMAND
*************************************************************
   USSCMD  CMD=LINES,FORMAT=BAL
   USSPARM PARM=DEFINED,REP=USE,VALUE=DEFINED
   USSPARM PARM=MAX
   USSPARM PARM=OWNER
   USSPARM PARM=SCOPE,DEFAULT=ALL
```

```
   USSPARM PARM=SPARE,REP=USE,VALUE=SPARE
   USSPARM PARM=USE
*************************************************************
* DISPLAY LMTBL COMMAND
*************************************************************
   USSCMD  CMD=DLMTBL,FORMAT=BAL
   USSPARM PARM=LU,REP=LUNAME
   USSPARM PARM=LUNAME
   USSPARM PARM=TYPE
*************************************************************
* DISPLAY LUGROUPS COMMAND
*************************************************************
   USSCMD  CMD=LUGROUPS,FORMAT=BAL
   USSPARM PARM=GROUP
   USSPARM PARM=SCOPE,DEFAULT=ONLY
   USSPARM PARM=MAX
*************************************************************
* DISPLAY MAJNODES COMMAND
*************************************************************
   USSCMD  CMD=MAJNODES,FORMAT=BAL
   USSPARM PARM=MAX
*************************************************************
* DISPLAY MODELS COMMAND
*************************************************************
   USSCMD  CMD=MODELS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ONLY
*************************************************************
* DISPLAY NCPSTOR COMMAND
*************************************************************
   USSCMD  CMD=NCPSTOR,FORMAT=BAL
   USSPARM PARM=ADDR
   USSPARM PARM=LENGTH,DEFAULT=32
   USSPARM PARM=STORAGE,DEFAULT=MAIN
*************************************************************
* DISPLAY NETSRVR COMMAND
*************************************************************
   USSCMD  CMD=NETSRVR,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY PATHS COMMAND
*************************************************************
   USSCMD  CMD=PATHS,FORMAT=BAL
*************************************************************
* DISPLAY PATHTAB COMMAND
*************************************************************
   USSCMD  CMD=PATHTAB,FORMAT=BAL
   USSPARM PARM=ADJSUB
   USSPARM PARM=DESTSUB
   USSPARM PARM=MAX
*************************************************************
* DISPLAY PENDING COMMAND
*************************************************************
   USSCMD  CMD=PENDING,FORMAT=BAL
   USSPARM PARM=MAX
*************************************************************
* DISPLAY ROUTE COMMAND
*************************************************************
   USSCMD  CMD=ROUTE,FORMAT=BAL
   USSPARM PARM=COSNAME
   USSPARM PARM=DESTSA,REP=DESTSUB
   USSPARM PARM=DESTSUB
   USSPARM PARM=ER
   USSPARM PARM=ORIGIN
   USSPARM PARM=TEST,DEFAULT=NO
   USSPARM PARM=VR
*************************************************************
* DISPLAY RSCLIST COMMAND
```

```
************************************************************
   USSCMD  CMD=RSCLIST,FORMAT=BAL
   USSPARM PARM=EXCLUDE
   USSPARM PARM=IDTYPE,DEFAULT=*
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY SATOAPPN COMMAND
************************************************************
   USSCMD  CMD=SATOAPPN,FORMAT=BAL
************************************************************
* DISPLAY SESSIONS COMMAND
************************************************************
   USSCMD  CMD=SESSIONS,FORMAT=BAL
   USSPARM PARM=LIST,DEFAULT=COUNT
   USSPARM PARM=LU1
   USSPARM PARM=LU2
   USSPARM PARM=MAX
   USSPARM PARM=PCID,REP=SID
   USSPARM PARM=PLU
   USSPARM PARM=SCOPE,DEFAULT=ALL
   USSPARM PARM=SID
   USSPARM PARM=SLU
************************************************************
* DISPLAY SRCHINFO COMMAND
************************************************************
   USSCMD  CMD=SRCHINFO,FORMAT=BAL
   USSPARM PARM=DLU
   USSPARM PARM=FROMCP
   USSPARM PARM=FROMSSCP
   USSPARM PARM=LIST,DEFAULT=SUMMARY
   USSPARM PARM=LU1
   USSPARM PARM=LU2
   USSPARM PARM=MAX
   USSPARM PARM=OLU
   USSPARM PARM=PCID,REP=SID
   USSPARM PARM=SID
   USSPARM PARM=TOCP
   USSPARM PARM=TOSSCP
   USSPARM PARM=TYPE,DEFAULT=ALL
************************************************************
* DISPLAY STATIONS COMMAND
************************************************************
   USSCMD  CMD=STATIONS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY STATS COMMAND
************************************************************
   USSCMD  CMD=STATS,FORMAT=BAL
   USSPARM PARM=NUM,REP=MAX
   USSPARM PARM=MAX
   USSPARM PARM=TYPE
************************************************************
* DISPLAY STORUSE COMMAND
************************************************************
   USSCMD  CMD=STORUSE,FORMAT=BAL
   USSPARM PARM=APPL
   USSPARM PARM=DSPNAME
   USSPARM PARM=JOBNAME
   USSPARM PARM=NUM,REP=MAX
   USSPARM PARM=MAX
   USSPARM PARM=POOL
************************************************************
* DISPLAY TABLE COMMAND
************************************************************
   USSCMD  CMD=DTABLE,FORMAT=BAL
```

```
   USSPARM PARM=SCOPE,DEFAULT=ONLY
   USSPARM PARM=MAX
*************************************************************
* DISPLAY TERMS COMMAND
*************************************************************
   USSCMD  CMD=TERMS,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY TGPS COMMAND
*************************************************************
   USSCMD  CMD=TGPS,FORMAT=BAL
   USSPARM PARM=MAX
*************************************************************
* DISPLAY TRACES COMMAND
*************************************************************
   USSCMD  CMD=TRACES,FORMAT=BAL
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=TYPE,DEFAULT=VTAM
*************************************************************
* DISPLAY TRL COMMAND
*************************************************************
   USSCMD  CMD=TRL,FORMAT=BAL
   USSPARM PARM=MAX
   USSPARM PARM=CONTROL,DEFAULT=ALL
   USSPARM PARM=XCFCP
   USSPARM PARM=TRLE
*************************************************************
* DISPLAY TSOUSER COMMAND
*************************************************************
   USSCMD  CMD=TSOUSER,FORMAT=BAL
*************************************************************
* DISPLAY USERVAR COMMAND
*************************************************************
   USSCMD  CMD=USERVAR,FORMAT=BAL
   USSPARM PARM=MAX
*************************************************************
* DISPLAY VTAMOPTS COMMAND
*************************************************************
   USSCMD  CMD=VTAMOPTS,FORMAT=BAL
   USSPARM PARM=COMP,REP=FORMAT,VALUE=COMPLETE
   USSPARM PARM=CUR,REP=FORMAT,VALUE=CURRENT
   USSPARM PARM=FUNCTION
   USSPARM PARM=MOD,REP=FORMAT,VALUE=MODIFIED
   USSPARM PARM=OPT,REP=OPTION
* DEFAULTS MUST FOLLOW VALUE ********************************
   USSPARM PARM=FORMAT,DEFAULT=CURRENT
   USSPARM PARM=OPTION,DEFAULT=*
*************************************************************
* MODIFY COMMAND
*************************************************************
   USSCMD  CMD=MODIFY,FORMAT=BAL
   USSPARM PARM=ALSLIST,REP=VERB,VALUE=ALSLIST
   USSPARM PARM=APINGDTP,REP=VERB,VALUE=FAPINGDT
   USSPARM PARM=ATTACH,REP=VERB,VALUE=ATTACH
   USSPARM PARM=CANCEL,REP=OPTION,VALUE=CANCEL
   USSPARM PARM=CDRM
   USSPARM PARM=CHKPT,REP=VERB,VALUE=CHKPT
   USSPARM PARM=CNOS,REP=VERB,VALUE=FCNOS
   USSPARM PARM=CNSL,REP=CNSL,VALUE=YES
   USSPARM PARM=COMPRESS,REP=VERB,VALUE=COMPRESS
   USSPARM PARM=DEFAULTS,REP=VERB,VALUE=DEFAULTS
   USSPARM PARM=DEFINE,REP=VERB,VALUE=FDEFINE
   USSPARM PARM=DETACH,REP=VERB,VALUE=DETACH
   USSPARM PARM=DIRECTRY,REP=VERB,VALUE=FDIRECT
   USSPARM PARM=DR,REP=VERB,VALUE=DR
   USSPARM PARM=DUMP,REP=VERB,VALUE=DUMP
```

```
   USSPARM PARM=DYNA,REP=OPTION,VALUE=DYNA
   USSPARM PARM=E,REP=SCOPE,VALUE=ALL
   USSPARM PARM=ENCR
   USSPARM PARM=EVERY,REP=SCOPE,VALUE=ALL
   USSPARM PARM=EXIT,REP=VERB,VALUE=FEXIT
   USSPARM PARM=F,REP=TYPE,VALUE=FORCE
   USSPARM PARM=FORCE,REP=TYPE,VALUE=FORCE
   USSPARM PARM=I,REP=TYPE,VALUE=IMMED
   USSPARM PARM=ID
   USSPARM PARM=IMMED,REP=TYPE,VALUE=IMMED
   USSPARM PARM=IMR,REP=VERB,VALUE=IMR
   USSPARM PARM=IOPD
   USSPARM PARM=LINEDEF,REP=VERB,VALUE=LINEDEF
   USSPARM PARM=LL2,REP=VERB,VALUE=LL2
   USSPARM PARM=LOAD,REP=VERB,VALUE=LOAD
   USSPARM PARM=MSG,REP=VERB,VALUE=MSG
   USSPARM PARM=N,REP=TYPE,VALUE=NORM
   USSPARM PARM=NCP,REP=VERB,VALUE=NCP
   USSPARM PARM=NEGPOLL
   USSPARM PARM=NOCNSL,REP=CNSL,VALUE=NO
   USSPARM PARM=NORM,REP=TYPE,VALUE=NORM
   USSPARM PARM=NOTNSTAT,REP=VERB,VALUE=NOTNSTAT
   USSPARM PARM=NOTRACE,REP=VERB,VALUE=NOTRACE
   USSPARM PARM=OPT,REP=OPTION
   USSPARM PARM=POLL
   USSPARM PARM=PROFILES,REP=VERB,VALUE=PROFILES
   USSPARM PARM=QUERY,REP=VERB,VALUE=QUERY
   USSPARM PARM=RESOURCE,REP=VERB,VALUE=RESOURCE
   USSPARM PARM=RMPO,REP=RMPO,VALUE=YES
   USSPARM PARM=RTP,REP=VERB,VALUE=RTP
   USSPARM PARM=SESSION
   USSPARM PARM=SUBTASK,REP=VERB,VALUE=SUBTASK
   USSPARM PARM=TABLE,REP=VERB,VALUE=TABLE
   USSPARM PARM=TGP,REP=VERB,VALUE=TGP
   USSPARM PARM=TGPS,REP=VERB,VALUE=TGP
   USSPARM PARM=TNSTAT,REP=VERB,VALUE=TNSTAT
   USSPARM PARM=TOPO,REP=VERB,VALUE=TOPO
   USSPARM PARM=TPRINT,REP=VERB,VALUE=TPRINT
   USSPARM PARM=TRACE,REP=VERB,VALUE=TRACE
   USSPARM PARM=TRACES,REP=VERB,VALUE=TRACE
   USSPARM PARM=TYPE
   USSPARM PARM=USERVAR,REP=VERB,VALUE=FUSERVAR
   USSPARM PARM=VTAMOPTS,REP=VERB,VALUE=FVTAMOPT
***************************************************************
* MODIFY ALSLIST COMMAND
***************************************************************
   USSCMD  CMD=ALSLIST
   USSPARM PARM=ACTION
   USSPARM PARM=NEW,REP=NEWALS
   USSPARM PARM=NEWALS
   USSPARM PARM=OLD,REP=OLDALS
   USSPARM PARM=OLDALS
***************************************************************
* MODIFY APINGDTP COMMAND
***************************************************************
   USSCMD  CMD=FAPINGDT,FORMAT=BAL
   USSPARM PARM=INSTANCE
***************************************************************
* MODIFY ATTACH COMMAND (VM AND VSE ONLY)
***************************************************************
   USSCMD  CMD=ATTACH,REP=SUBTASK
   USSPARM PARM=FUNCTION,DEFAULT=ATTACH
   USSPARM PARM=PARM                (VM ONLY)
   USSPARM PARM=SUPVR,DEFAULT=NO     (VM ONLY)
***************************************************************
* MODIFY CHKPT COMMAND
***************************************************************
```

```
   USSCMD  CMD=CHKPT,FORMAT=BAL
   USSPARM PARM=TYPE,DEFAULT=ALL
*************************************************************
* MODIFY COMPRESS COMMAND
*************************************************************
   USSCMD  CMD=COMPRESS,FORMAT=BAL
   USSPARM PARM=CMPAPPLI
   USSPARM PARM=CMPAPPLO
   USSPARM PARM=CMPVTAM
*************************************************************
* MODIFY CNOS COMMAND
*************************************************************
   USSCMD  CMD=FCNOS
   USSPARM PARM=CSECL,REP=CONVSECL
   USSPARM PARM=CONVSECL
   USSPARM PARM=DRL,REP=DRAINL,DEFAULT=NO
   USSPARM PARM=DRAINL,DEFAULT=NO
   USSPARM PARM=DRR,REP=DRAINR,DEFAULT=NO
   USSPARM PARM=DRAINR,DEFAULT=NO
   USSPARM PARM=LIMITS
   USSPARM PARM=LOG,REP=LOGMODE
   USSPARM PARM=LOGMODE
   USSPARM PARM=LU,REP=LUNAME
   USSPARM PARM=LUNAME
   USSPARM PARM=NBM,REP=NBRMODE,DEFAULT=ONE
   USSPARM PARM=NBRMODE,DEFAULT=ONE
   USSPARM PARM=RESP,DEFAULT=LOCAL
   USSPARM PARM=RSP,REP=RESP,DEFAULT=LOCAL
   USSPARM PARM=SNG,REP=SNGSESLU,DEFAULT=NO
   USSPARM PARM=SNGSESLU,DEFAULT=NO
*************************************************************
* MODIFY DEFAULTS COMMAND
*************************************************************
   USSCMD  CMD=DEFAULTS
   USSPARM PARM=DLOG,REP=DLOGMOD
   USSPARM PARM=DLOGMOD
*************************************************************
* MODIFY DEFINE COMMAND
*************************************************************
   USSCMD  CMD=FDEFINE
   USSPARM PARM=AUTO,REP=AUTOSES
   USSPARM PARM=AUTOSES
   USSPARM PARM=DEL,REP=DELETE,DEFAULT=NALLOW
   USSPARM PARM=DELETE,DEFAULT=NALLOW
   USSPARM PARM=DDRL,REP=DDRAINL
   USSPARM PARM=DDRAINL
   USSPARM PARM=DLIMITS
   USSPARM PARM=DRSP,REP=DRESPL
   USSPARM PARM=DRESPL
   USSPARM PARM=LOG,REP=LOGMODE
   USSPARM PARM=LOGMODE
   USSPARM PARM=LU,REP=LUNAME
   USSPARM PARM=LUNAME
*************************************************************
* MODIFY DETACH COMMAND (VM AND VSE ONLY)
*************************************************************
   USSCMD  CMD=DETACH,REP=SUBTASK
   USSPARM PARM=FUNCTION,DEFAULT=DETACH
*************************************************************
* MODIFY DIRECTRY COMMAND
*************************************************************
   USSCMD  CMD=FDIRECT
   USSPARM PARM=CPNAME
   USSPARM PARM=DELETE,REP=FUNCTION,VALUE=DELETE
   USSPARM PARM=FUNCTION
   USSPARM PARM=NETSRVR
   USSPARM PARM=UPDATE,REP=FUNCTION,VALUE=UPDATE
```

```
*************************************************************
* MODIFY DR COMMAND
*************************************************************
   USSCMD  CMD=DR
   USSPARM PARM=ACT,REP=ACTIVATE
   USSPARM PARM=ACTIVATE
   USSPARM PARM=ADDR
   USSPARM PARM=FROM
   USSPARM PARM=TO
*************************************************************
* MODIFY DUMP COMMAND
*************************************************************
   USSCMD  CMD=DUMP
   USSPARM PARM=ACTION,DEFAULT=COMP
   USSPARM PARM=DS,REP=DUMPDS
   USSPARM PARM=DST,REP=DUMPSTA
   USSPARM PARM=DUMPDS
   USSPARM PARM=DUMPSTA
   USSPARM PARM=OPT,REP=OPTION,DEFAULT=STATIC
   USSPARM PARM=OPTION,DEFAULT=STATIC
   USSPARM PARM=RMPO,DEFAULT=NO
   USSPARM PARM=TRANS,REP=ACTION,VALUE=TRANSFER
   USSPARM PARM=TYPE,DEFAULT=NCP
*************************************************************
* MODIFY EXIT COMMAND
*************************************************************
   USSCMD  CMD=FEXIT
   USSPARM PARM=MOD,REP=MODULE
   USSPARM PARM=MODULE
   USSPARM PARM=OPTION
   USSPARM PARM=PARMS
*************************************************************
* MODIFY IMR COMMAND
*************************************************************
   USSCMD  CMD=IMR
   USSPARM PARM=OPT,REP=OPTION,DEFAULT=ACT
   USSPARM PARM=OPTION,DEFAULT=ACT
   USSPARM PARM=RECLIM,DEFAULT=10
*************************************************************
* MODIFY IOPD COMMAND
*************************************************************
   USSCMD  CMD=IOPD
   USSPARM PARM=IOINT
*************************************************************
* MODIFY LINEDEF COMMAND
*************************************************************
   USSCMD  CMD=LINEDEF
   USSPARM PARM=DEFINED,REP=USE,VALUE=DEFINED
   USSPARM PARM=SPARE,REP=USE,VALUE=SPARE
   USSPARM PARM=USE
*************************************************************
* MODIFY LL2 COMMAND
*************************************************************
   USSCMD  CMD=LL2
   USSPARM PARM=CANCEL,REP=OPTION,VALUE=CANCEL
   USSPARM PARM=CONT,REP=OPTION,VALUE=CONT
   USSPARM PARM=DATA
   USSPARM PARM=NFRAMES,DEFAULT=1
   USSPARM PARM=NTRANS,DEFAULT=10
*************************************************************
* MODIFY LOAD COMMAND
*************************************************************
   USSCMD  CMD=LOAD,FORMAT=BAL
   USSPARM PARM=ACTION,DEFAULT=ADD
   USSPARM PARM=LM,REP=LOADMOD
   USSPARM PARM=LOADMOD
   USSPARM PARM=NEWN,REP=NEWNAME
```

```
   USSPARM PARM=NEWNAME
   USSPARM PARM=IPLT,REP=IPLTIME
   USSPARM PARM=IPLTIME
   USSPARM PARM=NOTIFY
   USSPARM PARM=NT,REP=NOTIFY
*************************************************************
* MODIFY MSG COMMAND (VSE ONLY)
*************************************************************
   USSCMD  CMD=MSG,REP=SUBTASK
   USSPARM PARM=FUNCTION,DEFAULT=MSG
*************************************************************
* MODIFY NCP COMMAND
*************************************************************
   USSCMD  CMD=NCP,FORMAT=BAL
   USSPARM PARM=COM,REP=COMMAND
   USSPARM PARM=CMD,REP=COMMAND
   USSPARM PARM=COMMAND,DEFAULT=TRSWITCH
   USSPARM PARM=RES,REP=RESNM
   USSPARM PARM=NAME,REP=RESNM
   USSPARM PARM=RESNAME,REP=RESNM
   USSPARM PARM=RESNM
*************************************************************
* MODIFY NOTNSTAT COMMAND
*************************************************************
   USSCMD  CMD=NOTNSTAT
*************************************************************
* MODIFY NOTRACE COMMAND
*************************************************************
   USSCMD  CMD=NOTRACE
   USSPARM PARM=ALS,REP=ALSNAME
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=LINE
   USSPARM PARM=MODE
   USSPARM PARM=NONE,REP=SCOPE,VALUE=ONLY
   USSPARM PARM=PU
   USSPARM PARM=SCOPE,DEFAULT=ONLY
*************************************************************
* MODIFY PROFILES COMMAND
*************************************************************
   USSCMD  CMD=PROFILES,FORMAT=BAL
*************************************************************
* MODIFY QUERY COMMAND (POA ONLY)
*************************************************************
   USSCMD  CMD=QUERY,FORMAT=BAL
   USSPARM PARM=CONT,REP=CONTINUE
   USSPARM PARM=NO,REP=CONTINUE,VALUE=NO
   USSPARM PARM=RESET,REP=CONTINUE,VALUE=RESET
   USSPARM PARM=YES,REP=CONTINUE,VALUE=YES
   USSPARM PARM=CONTINUE,DEFAULT=NO
   USSPARM PARM=V,REP=VECTORS
   USSPARM PARM=VECTOR,REP=VECTORS
   USSPARM PARM=VECTORS
*************************************************************
* MODIFY RESOURCE COMMAND
*************************************************************
   USSCMD  CMD=RESOURCE
   USSPARM PARM=ACTION
   USSPARM PARM=ADJLIST
   USSPARM PARM=DLOG,REP=DLOGMOD
   USSPARM PARM=DLOGMOD
*************************************************************
* MODIFY RTP COMMAND
*************************************************************
   USSCMD  CMD=RTP,FORMAT=BAL
   USSPARM PARM=ID
*************************************************************
* MODIFY SUBTASK COMMAND (VM AND VSE ONLY)
```

```
*************************************************************
   USSCMD  CMD=SUBTASK
   USSPARM PARM=FUNCTION,DEFAULT=ATTACH
   USSPARM PARM=PARM                (VM ONLY)
   USSPARM PARM=SUPVR,DEFAULT=NO    (VM ONLY)
*************************************************************
* MODIFY TABLE COMMAND
*************************************************************
   USSCMD  CMD=TABLE,FORMAT=BAL
   USSPARM PARM=NET,REP=NETID
   USSPARM PARM=NETID
   USSPARM PARM=NEW,REP=NEWTAB
   USSPARM PARM=NEWTAB
   USSPARM PARM=OLD,REP=OLDTAB
   USSPARM PARM=OLDTAB
   USSPARM PARM=OPTION
   USSPARM PARM=ORIGIN
*************************************************************
* MODIFY TGP COMMAND
*************************************************************
   USSCMD  CMD=TGP,FORMAT=BAL
   USSPARM PARM=TGN
   USSPARM PARM=TGPNAME
*************************************************************
* MODIFY TOPO COMMAND
*************************************************************
   USSCMD  CMD=TOPO,FORMAT=BAL
   USSPARM PARM=DELETE,REP=FUNCTION,VALUE=DELETE
   USSPARM PARM=DEST
   USSPARM PARM=FUNCTION
   USSPARM PARM=ORIG
   USSPARM PARM=TGN
*************************************************************
* MODIFY TNSTAT COMMAND
*************************************************************
   USSCMD  CMD=TNSTAT
   USSPARM PARM=CNSL,REP=CNSL,VALUE=YES
   USSPARM PARM=CNSL,DEFAULT=NO
   USSPARM PARM=NOCNSL,REP=CNSL,VALUE=NO
   USSPARM PARM=TIME
*************************************************************
* MODIFY TPRINT COMMAND (VSE ONLY)
*************************************************************
   USSCMD  CMD=TPRINT,REP=SUBTASK
   USSPARM PARM=FUNCTION,DEFAULT=ATTACH
   USSPARM PARM=ID,DEFAULT=TPRINT
*************************************************************
* MODIFY TRACE COMMAND
*************************************************************
   USSCMD  CMD=TRACE
   USSPARM PARM=ALS,REP=ALSNAME
   USSPARM PARM=AMT,REP=AMOUNT
   USSPARM PARM=AMOUNT,DEFAULT=PARTIAL
   USSPARM PARM=COUNT,DEFAULT=ALL
   USSPARM PARM=FRAMES,DEFAULT=DATA
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=LINE
   USSPARM PARM=MODE
   USSPARM PARM=NONE,REP=SCOPE,VALUE=ONLY
   USSPARM PARM=PU
   USSPARM PARM=SAVE,REP=SAVE,VALUE=YES
   USSPARM PARM=SAVE,DEFAULT=NO
   USSPARM PARM=SCOPE,DEFAULT=ONLY
   USSPARM PARM=SIZE
   USSPARM PARM=TRACEPT
*************************************************************
* MODIFY USERVAR COMMAND
```

```
*************************************************************
   USSCMD  CMD=FUSERVAR,FORMAT=BAL
   USSPARM PARM=APPC,DEFAULT=NO
   USSPARM PARM=OPT,REP=OPTION,DEFAULT=UPDATE
   USSPARM PARM=OPTION,DEFAULT=UPDATE
   USSPARM PARM=TYPE,DEFAULT=DYNAMIC
   USSPARM PARM=UVEXIT,DEFAULT=NO
   USSPARM PARM=VAL,REP=VALUE
   USSPARM PARM=VALUE
*************************************************************
* MODIFY VTAMOPTS COMMAND
*************************************************************
   USSCMD  CMD=FVTAMOPT,FORMAT=BAL
   USSPARM PARM=CSALIMIT
   USSPARM PARM=IOPD
   USSPARM PARM=IOPURGE
   USSPARM PARM=MSGMOD
   USSPARM PARM=PPOLOG
   USSPARM PARM=SGALIMIT
   USSPARM PARM=SUPP
*************************************************************
* VARY COMMANDS
*************************************************************
   USSCMD  CMD=VARY,FORMAT=BAL
   USSPARM PARM=ACQ,REP=ACQ
   USSPARM PARM=ACT,REP=VERB,VALUE=ACT
   USSPARM PARM=ANS
   USSPARM PARM=DIAL,REP=VERB,VALUE=DIAL
   USSPARM PARM=DRDS,REP=VERB,VALUE=DRDS
   USSPARM PARM=GID
   USSPARM PARM=HANGUP,REP=VERB,VALUE=HANGUP
   USSPARM PARM=ID
   USSPARM PARM=INACT,REP=VERB,VALUE=INACT
   USSPARM PARM=INOP,REP=VERB,VALUE=INOP
   USSPARM PARM=LM,REP=LOADMOD
   USSPARM PARM=LOADMOD
   USSPARM PARM=LOGMODE
   USSPARM PARM=LOGON
   USSPARM PARM=NOLOGON
   USSPARM PARM=P,REP=PUSUB
   USSPARM PARM=PATH
   USSPARM PARM=PID
   USSPARM PARM=PUSUB
   USSPARM PARM=REL,REP=VERB,VALUE=REL
   USSPARM PARM=TERM,REP=VERB,VALUE=TERM
*************************************************************
* VARY ACT
*************************************************************
   USSCMD  CMD=ACT,FORMAT=BAL
   USSPARM PARM=ALL,REP=SCOPE,VALUE=ALL
   USSPARM PARM=COMP,REP=SCOPE,VALUE=COMP
   USSPARM PARM=CPCP
   USSPARM PARM=DL,REP=DUMPLOAD
   USSPARM PARM=DST,REP=DUMPSTA
   USSPARM PARM=DUMPLOAD
   USSPARM PARM=DUMPSTA
   USSPARM PARM=EXT,REP=LOADFROM,VALUE=EXTERNAL
   USSPARM PARM=HOST,REP=LOADFROM,VALUE=HOST
   USSPARM PARM=HPR
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=LF,REP=LOADFROM
   USSPARM PARM=LOAD,DEFAULT=U
   USSPARM PARM=LOADFROM
   USSPARM PARM=LST,REP=LOADSTA
   USSPARM PARM=LOADSTA
   USSPARM PARM=LOG,REP=LOGMODE
   USSPARM PARM=LOGMODE
```

```
   USSPARM PARM=NEWPATH
   USSPARM PARM=ONLY,REP=SCOPE,VALUE=ONLY
   USSPARM PARM=OWNER
   USSPARM PARM=PUSUB
   USSPARM PARM=RNAME
   USSPARM PARM=SM,REP=SAVEMOD
   USSPARM PARM=SAVEMOD
   USSPARM PARM=SCOPE
   USSPARM PARM=SYNTAX,REP=SCOPE,VALUE=SYNTAX
   USSPARM PARM=U
   USSPARM PARM=UPDATE,DEFAULT=IMPLICIT
   USSPARM PARM=WARM,REP=WARM
*************************************************************
* VARY DIAL COMMAND
*************************************************************
   USSCMD  CMD=DIAL,FORMAT=BAL
   USSPARM PARM=CPCP
*************************************************************
* VARY DRDS
*************************************************************
   USSCMD  CMD=DRDS
*************************************************************
* VARY INACT
*************************************************************
   USSCMD  CMD=INACT
   USSPARM PARM=CDLINK
   USSPARM PARM=DELETE,REP=DELETE,VALUE=YES
   USSPARM PARM=F,REP=TYPE,VALUE=FORCE
   USSPARM PARM=FINAL,REP=FINAL,VALUE=YES
   USSPARM PARM=FINAL,DEFAULT=NO
   USSPARM PARM=FORCE,REP=TYPE,VALUE=FORCE
   USSPARM PARM=G,REP=TYPE,VALUE=GIVEBACK
   USSPARM PARM=GIVEBACK,REP=TYPE,VALUE=GIVEBACK
   USSPARM PARM=I,REP=TYPE,VALUE=IMMED
   USSPARM PARM=IDTYPE
   USSPARM PARM=IMMED,REP=TYPE,VALUE=IMMED
   USSPARM PARM=R,REP=TYPE,VALUE=REACT
   USSPARM PARM=REACT,REP=TYPE,VALUE=REACT
   USSPARM PARM=RMPO,REP=RMPO,VALUE=YES
   USSPARM PARM=RMPO,DEFAULT=NO
   USSPARM PARM=S,REP=SAVESESS
   USSPARM PARM=SAVESESS
   USSPARM PARM=TYPE
   USSPARM PARM=U,REP=TYPE,VALUE=UNCOND
   USSPARM PARM=UNCOND,REP=TYPE,VALUE=UNCOND
*************************************************************
* VARY INOP COMMAND
*************************************************************
   USSCMD  CMD=INOP
*************************************************************
* VARY HANGUP COMMAND
*************************************************************
   USSCMD  CMD=HANGUP
*************************************************************
* VARY REL COMMAND
*************************************************************
   USSCMD  CMD=REL,FORMAT=BAL
   USSPARM PARM=CDLINK,DEFAULT=ACT
   USSPARM PARM=FINAL,REP=FINAL,VALUE=YES
   USSPARM PARM=FINAL,DEFAULT=NO
   USSPARM PARM=G,REP=TYPE,VALUE=GIVEBACK
   USSPARM PARM=GIVEBACK,REP=TYPE,VALUE=GIVEBACK
   USSPARM PARM=I,REP=TYPE,VALUE=IMMED
   USSPARM PARM=IMMED,REP=TYPE,VALUE=IMMED
   USSPARM PARM=OWNER
*************************************************************
* VARY TERM COMMAND (SESSION SERVICES)
```

Operation-Level USS Table

```
*************************************************************
   USSCMD  CMD=TERM,FORMAT=BAL
   USSPARM PARM=LU1
   USSPARM PARM=LU2
   USSPARM PARM=NOTIFY,DEFAULT=YES
   USSPARM PARM=PLU
   USSPARM PARM=SCOPE,DEFAULT=ACT
   USSPARM PARM=SID
   USSPARM PARM=SLU
   USSPARM PARM=TYPE,DEFAULT=UNCOND
   USSEND
*  END OF ISTINCNO USS DEFINITION TABLE
   END
```

APPN Class-of-Service (COS) Definitions (COSAPPN)

Two sets of IBM-supplied APPN class-of-service definitions are available:

- COSAPPN
- ISTACST2

COSAPPN

```
CPSVCMG  APPNCOS  PRIORITY=NETWORK          transmission priority
         LINEROW  WEIGHT=30,                line row weight            *
               NUMBER=1,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(4M,MAXIMUM),       line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=1,                 node row number            *
               WEIGHT=5,                    node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,31)              route addition resistance
         LINEROW  WEIGHT=60,                line row weight            *
               NUMBER=2,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(56000,MAXIMUM),    line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=2,                 node row number            *
               WEIGHT=10,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,63)              route addition resistance
         LINEROW  WEIGHT=90,                line row weight            *
               NUMBER=3,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(19200,MAXIMUM),    line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=3,                 node row number            *
               WEIGHT=20,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,95)              route addition resistance
```

```
LINEROW  WEIGHT=120,                line row weight            *
      NUMBER=4,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(9600,MAXIMUM),      line speed                 *
      COSTTIME=(0,0),               cost per connect time      *
      COSTBYTE=(0,0),               cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=4,                  node row number            *
      WEIGHT=40,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,127)              route addition resistance
LINEROW  WEIGHT=150,                line row weight            *
      NUMBER=5,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(19200,MAXIMUM),     line speed                 *
      COSTTIME=(0,0),               cost per connect time      *
      COSTBYTE=(0,0),               cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),      propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=5,                  node row number            *
      WEIGHT=60,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,159)              route addition resistance
LINEROW  WEIGHT=180,                line row weight            *
      NUMBER=6,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(9600,MAXIMUM),      line speed                 *
      COSTTIME=(0,128),             cost per connect time      *
      COSTBYTE=(0,128),             cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),      propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=6,                  node row number            *
      WEIGHT=80,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,191)              route addition resistance
LINEROW  WEIGHT=210,                line row weight            *
      NUMBER=7,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(4800,MAXIMUM),      line speed                 *
      COSTTIME=(0,196),             cost per connect time      *
      COSTBYTE=(0,196),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=7,                  node row number            *
      WEIGHT=120,                   node row weight            *
      CONGEST=(LOW,HIGH),           congestion                 *
      ROUTERES=(0,223)              route addition resistance
LINEROW  WEIGHT=240,                line row weight            *
      NUMBER=8,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(MINIMUM,MAXIMUM),   line speed                 *
      COSTTIME=(0,255),             cost per connect time      *
      COSTBYTE=(0,255),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=8,                  node row number            *
```

```
                WEIGHT=160,                 node row weight           *
                CONGEST=(LOW,HIGH),         congestion                *
                ROUTERES=(0,255)            route addition resistance
*
SNASVCMG APPNCOS  PRIORITY=NETWORK          transmission priority
         LINEROW  WEIGHT=30,                line row weight           *
                NUMBER=1,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(4M,MAXIMUM),      line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,NEGLIGIB),  propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
         NODEROW  NUMBER=1,                 node row number           *
                WEIGHT=5,                   node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,31)             route addition resistance
         LINEROW  WEIGHT=60,                line row weight           *
                NUMBER=2,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(56000,MAXIMUM),   line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,TERRESTR),  propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
         NODEROW  NUMBER=2,                 node row number           *
                WEIGHT=10,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,63)             route addition resistance
         LINEROW  WEIGHT=90,                line row weight           *
                NUMBER=3,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(19200,MAXIMUM),   line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,TERRESTR),  propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
         NODEROW  NUMBER=3,                 node row number           *
                WEIGHT=20,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,95)             route addition resistance
         LINEROW  WEIGHT=120,               line row weight           *
                NUMBER=4,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(9600,MAXIMUM),    line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,TERRESTR),  propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
         NODEROW  NUMBER=4,                 node row number           *
                WEIGHT=40,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,127)            route addition resistance
         LINEROW  WEIGHT=150,               line row weight           *
                NUMBER=5,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(19200,MAXIMUM),   line speed                *
```

```
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,PACKET),     propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=5,                 node row number            *
                WEIGHT=60,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,159)             route addition resistance
          LINEROW  WEIGHT=180,               line row weight            *
                NUMBER=6,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(9600,MAXIMUM),     line speed                 *
                COSTTIME=(0,128),            cost per connect time      *
                COSTBYTE=(0,128),            cost per byte transmitted  *
                PDELAY=(MINIMUM,PACKET),     propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=6,                 node row number            *
                WEIGHT=80,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,191)             route addition resistance
          LINEROW  WEIGHT=210,               line row weight            *
                NUMBER=7,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4800,MAXIMUM),     line speed                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=7,                 node row number            *
                WEIGHT=120,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,223)             route addition resistance
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=8,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#CONNECT APPNCOS  PRIORITY=MEDIUM            transmission priority
          LINEROW  WEIGHT=30,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4M,MAXIMUM),       line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
```

```
      NUMBER=2,                    line row number           *
      UPARM1=(0,255),              user defined char 1       *
      UPARM2=(0,255),              user defined char 2       *
      UPARM3=(0,255),              user defined char 3       *
      CAPACITY=(56000,MAXIMUM),    line speed                *
      COSTTIME=(0,0),              cost per connect time     *
      COSTBYTE=(0,0),              cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=2,                 node row number           *
      WEIGHT=10,                   node row weight           *
      CONGEST=(LOW,LOW),           congestion                *
      ROUTERES=(0,63)              route addition resistance
LINEROW  WEIGHT=90,                line row weight           *
      NUMBER=3,                    line row number           *
      UPARM1=(0,255),              user defined char 1       *
      UPARM2=(0,255),              user defined char 2       *
      UPARM3=(0,255),              user defined char 3       *
      CAPACITY=(19200,MAXIMUM),    line speed                *
      COSTTIME=(0,0),              cost per connect time     *
      COSTBYTE=(0,0),              cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=3,                 node row number           *
      WEIGHT=20,                   node row weight           *
      CONGEST=(LOW,LOW),           congestion                *
      ROUTERES=(0,95)              route addition resistance
LINEROW  WEIGHT=120,               line row weight           *
      NUMBER=4,                    line row number           *
      UPARM1=(0,255),              user defined char 1       *
      UPARM2=(0,255),              user defined char 2       *
      UPARM3=(0,255),              user defined char 3       *
      CAPACITY=(9600,MAXIMUM),     line speed                *
      COSTTIME=(0,0),              cost per connect time     *
      COSTBYTE=(0,0),              cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=4,                 node row number           *
      WEIGHT=40,                   node row weight           *
      CONGEST=(LOW,LOW),           congestion                *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=150,               line row weight           *
      NUMBER=5,                    line row number           *
      UPARM1=(0,255),              user defined char 1       *
      UPARM2=(0,255),              user defined char 2       *
      UPARM3=(0,255),              user defined char 3       *
      CAPACITY=(19200,MAXIMUM),    line speed                *
      COSTTIME=(0,0),              cost per connect time     *
      COSTBYTE=(0,0),              cost per byte transmitted *
      PDELAY=(MINIMUM,PACKET),     propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=5,                 node row number           *
      WEIGHT=60,                   node row weight           *
      CONGEST=(LOW,LOW),           congestion                *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=180,               line row weight           *
      NUMBER=6,                    line row number           *
      UPARM1=(0,255),              user defined char 1       *
      UPARM2=(0,255),              user defined char 2       *
      UPARM3=(0,255),              user defined char 3       *
      CAPACITY=(9600,MAXIMUM),     line speed                *
      COSTTIME=(0,128),            cost per connect time     *
      COSTBYTE=(0,128),            cost per byte transmitted *
      PDELAY=(MINIMUM,PACKET),     propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=6,                 node row number           *
      WEIGHT=80,                   node row weight           *
```

```
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,191)             route addition resistance
          LINEROW  WEIGHT=210,               line row weight            *
                NUMBER=7,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4800,MAXIMUM),     line speed                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=7,                 node row number            *
                WEIGHT=120,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,223)             route addition resistance
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=8,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#INTER    APPNCOS  PRIORITY=HIGH             transmission priority
          LINEROW  WEIGHT=30,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4M,MAXIMUM),       line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(56000,MAXIMUM),    line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=2,                 node row number            *
                WEIGHT=10,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,63)              route addition resistance
          LINEROW  WEIGHT=90,                line row weight            *
                NUMBER=3,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(56000,MAXIMUM),    line speed                 *
                COSTTIME=(0,128),            cost per connect time      *
```

```
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=3,                 node row number            *
      WEIGHT=20,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,95)              route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=4,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=150,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=210,               line row weight            *
      NUMBER=7,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,196),            cost per connect time      *
      COSTBYTE=(0,196),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=7,                 node row number            *
      WEIGHT=120,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,223)             route addition resistance
LINEROW  WEIGHT=240,               line row weight            *
      NUMBER=8,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
```

```
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#INTERSC APPNCOS  PRIORITY=HIGH              transmission priority
          LINEROW  WEIGHT=30,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4M,MAXIMUM),       line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(56000,MAXIMUM),    line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=2,                 node row number            *
                WEIGHT=10,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,63)              route addition resistance
          LINEROW  WEIGHT=90,                line row weight            *
                NUMBER=3,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(56000,MAXIMUM),    line speed                 *
                COSTTIME=(0,128),            cost per connect time      *
                COSTBYTE=(0,128),            cost per byte transmitted  *
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=3,                 node row number            *
                WEIGHT=20,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,95)              route addition resistance
          LINEROW  WEIGHT=120,               line row weight            *
                NUMBER=4,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(19200,MAXIMUM),    line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=4,                 node row number            *
                WEIGHT=40,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
```

```
                ROUTERES=(0,127)            route addition resistance
          LINEROW  WEIGHT=150,              line row weight           *
                NUMBER=5,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(19200,MAXIMUM),   line speed                *
                COSTTIME=(0,128),           cost per connect time     *
                COSTBYTE=(0,128),           cost per byte transmitted *
                PDELAY=(MINIMUM,PACKET),    propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=5,                node row number           *
                WEIGHT=60,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,159)            route addition resistance
          LINEROW  WEIGHT=180,              line row weight           *
                NUMBER=6,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(9600,MAXIMUM),    line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,PACKET),    propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=6,                node row number           *
                WEIGHT=80,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,191)            route addition resistance
          LINEROW  WEIGHT=210,              line row weight           *
                NUMBER=7,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(9600,MAXIMUM),    line speed                *
                COSTTIME=(0,196),           cost per connect time     *
                COSTBYTE=(0,196),           cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=7,                node row number           *
                WEIGHT=120,                 node row weight           *
                CONGEST=(LOW,HIGH),         congestion                *
                ROUTERES=(0,223)            route addition resistance
          LINEROW  WEIGHT=240,              line row weight           *
                NUMBER=8,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(MINIMUM,MAXIMUM), line speed                *
                COSTTIME=(0,255),           cost per connect time     *
                COSTBYTE=(0,255),           cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=8,                node row number           *
                WEIGHT=160,                 node row weight           *
                CONGEST=(LOW,HIGH),         congestion                *
                ROUTERES=(0,255)            route addition resistance
*
#BATCH    APPNCOS  PRIORITY=LOW             transmission priority
          LINEROW  WEIGHT=30,               line row weight           *
                NUMBER=1,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(56000,MAXIMUM),   line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
```

```
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=1,                 node row number            *
      WEIGHT=5,                    node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,31)              route addition resistance
LINEROW  WEIGHT=60,                line row weight            *
      NUMBER=2,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=2,                 node row number            *
      WEIGHT=10,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,63)              route addition resistance
LINEROW  WEIGHT=90,                line row weight            *
      NUMBER=3,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=3,                 node row number            *
      WEIGHT=20,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,95)              route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=4,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=150,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             ROUTE ADDITION RESISTANCE
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
```

```
                CAPACITY=(9600,MAXIMUM),     line speed                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=6,                 node row number            *
                WEIGHT=80,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,191)             route addition resistance
          LINEROW  WEIGHT=210,               line row weight            *
                NUMBER=7,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4800,MAXIMUM),     line speed                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=7,                 node row number            *
                WEIGHT=120,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,223)             route addition resistance
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=8,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#BATCHSC APPNCOS  PRIORITY=LOW               transmission priority
          LINEROW  WEIGHT=30,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(56000,MAXIMUM),    line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(19200,MAXIMUM),    line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=2,                 node row number            *
                WEIGHT=10,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,63)              route addition resistance
```

```
LINEROW  WEIGHT=90,                 line row weight            *
      NUMBER=3,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(19200,MAXIMUM),     line speed                 *
      COSTTIME=(0,128),             cost per connect time      *
      COSTBYTE=(0,128),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)     security level for TG
NODEROW  NUMBER=3,                  node row number            *
      WEIGHT=20,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,95)               route addition resistance
LINEROW  WEIGHT=120,                line row weight            *
      NUMBER=4,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(9600,MAXIMUM),      line speed                 *
      COSTTIME=(0,0),               cost per connect time      *
      COSTBYTE=(0,0),               cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)     security level for TG
NODEROW  NUMBER=4,                  node row number            *
      WEIGHT=40,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,127)              route addition resistance
LINEROW  WEIGHT=150,                line row weight            *
      NUMBER=5,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(9600,MAXIMUM),      line speed                 *
      COSTTIME=(0,128),             cost per connect time      *
      COSTBYTE=(0,128),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)     security level for TG
NODEROW  NUMBER=5,                  node row number            *
      WEIGHT=60,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,159)              route addition resistance
LINEROW  WEIGHT=180,                line row weight            *
      NUMBER=6,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(9600,MAXIMUM),      line speed                 *
      COSTTIME=(0,196),             cost per connect time      *
      COSTBYTE=(0,196),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)     security level for TG
NODEROW  NUMBER=6,                  node row number            *
      WEIGHT=80,                    node row weight            *
      CONGEST=(LOW,LOW),            congestion                 *
      ROUTERES=(0,191)              route addition resistance
LINEROW  WEIGHT=210,                line row weight            *
      NUMBER=7,                     line row number            *
      UPARM1=(0,255),               user defined char 1        *
      UPARM2=(0,255),               user defined char 2        *
      UPARM3=(0,255),               user defined char 3        *
      CAPACITY=(4800,MAXIMUM),      line speed                 *
      COSTTIME=(0,196),             cost per connect time      *
      COSTBYTE=(0,196),             cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)     security level for TG
NODEROW  NUMBER=7,                  node row number            *
```

```
                    WEIGHT=120,                 node row weight           *
                    CONGEST=(LOW,HIGH),         congestion                *
                    ROUTERES=(0,223)            route addition resistance
              LINEROW  WEIGHT=240,              line row weight           *
                    NUMBER=8,                   line row number           *
                    UPARM1=(0,255),             user defined char 1       *
                    UPARM2=(0,255),             user defined char 2       *
                    UPARM3=(0,255),             user defined char 3       *
                    CAPACITY=(MINIMUM,MAXIMUM), line speed                *
                    COSTTIME=(0,255),           cost per connect time     *
                    COSTBYTE=(0,255),           cost per byte transmitted *
                    PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                    SECURITY=(PUBLIC,MAXIMUM)   security level for TG
              NODEROW  NUMBER=8,                node row number           *
                    WEIGHT=160,                 node row weight           *
                    CONGEST=(LOW,HIGH),         congestion                *
                    ROUTERES=(0,255)            route addition resistance
```

ISTACST2

```
        CPSVCMG  APPNCOS  PRIORITY=NETWORK       transmission priority
              LINEROW  WEIGHT=30,               line row weight           *
                    NUMBER=1,                   line row number           *
                    UPARM1=(0,255),             user defined char 1       *
                    UPARM2=(0,255),             user defined char 2       *
                    UPARM3=(0,255),             user defined char 3       *
                    CAPACITY=(4M,MAXIMUM),      line speed                *
                    COSTTIME=(0,0),             cost per connect time     *
                    COSTBYTE=(0,0),             cost per byte transmitted *
                    PDELAY=(MINIMUM,NEGLIGIB),  propagation delay         *
                    SECURITY=(UNSECURE,MAXIMUM) security level for TG
              NODEROW  NUMBER=1,                node row number           *
                    WEIGHT=5,                   node row weight           *
                    CONGEST=(LOW,LOW),          congestion                *
                    ROUTERES=(0,31)             route addition resistance
              LINEROW  WEIGHT=60,               line row weight           *
                    NUMBER=2,                   line row number           *
                    UPARM1=(0,255),             user defined char 1       *
                    UPARM2=(0,255),             user defined char 2       *
                    UPARM3=(0,255),             user defined char 3       *
                    CAPACITY=(56000,MAXIMUM),   line speed                *
                    COSTTIME=(0,0),             cost per connect time     *
                    COSTBYTE=(0,0),             cost per byte transmitted *
                    PDELAY=(MINIMUM,TERRESTR),  propagation delay         *
                    SECURITY=(UNSECURE,MAXIMUM) security level for TG
              NODEROW  NUMBER=2,                node row number           *
                    WEIGHT=10,                  node row weight           *
                    CONGEST=(LOW,LOW),          congestion                *
                    ROUTERES=(0,63)             route addition resistance
              LINEROW  WEIGHT=90,               line row weight           *
                    NUMBER=3,                   line row number           *
                    UPARM1=(0,255),             user defined char 1       *
                    UPARM2=(0,255),             user defined char 2       *
                    UPARM3=(0,255),             user defined char 3       *
                    CAPACITY=(19200,MAXIMUM),   line speed                *
                    COSTTIME=(0,0),             cost per connect time     *
                    COSTBYTE=(0,0),             cost per byte transmitted *
                    PDELAY=(MINIMUM,TERRESTR),  propagation delay         *
                    SECURITY=(UNSECURE,MAXIMUM) security level for TG
              NODEROW  NUMBER=3,                node row number           *
                    WEIGHT=20,                  node row weight           *
                    CONGEST=(LOW,LOW),          congestion                *
                    ROUTERES=(0,95)             route addition resistance
              LINEROW  WEIGHT=120,              line row weight           *
                    NUMBER=4,                   line row number           *
                    UPARM1=(0,255),             user defined char 1       *
                    UPARM2=(0,255),             user defined char 2       *
```

```
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=150,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    LINE SPEED                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     LINE SPEED                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=210,               line row weight            *
      NUMBER=7,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(4800,MAXIMUM),     LINE SPEED                 *
      COSTTIME=(0,196),            cost per connect time      *
      COSTBYTE=(0,196),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=7,                 node row number            *
      WEIGHT=120,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,223)             route addition resistance
LINEROW  WEIGHT=240,               line row weight            *
      NUMBER=8,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
      COSTTIME=(0,255),            cost per connect time      *
      COSTBYTE=(0,255),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=8,                 node row number            *
      WEIGHT=160,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,255)             route addition resistance
*
```

APPN COS Definitions

```
SNASVCMG APPNCOS  PRIORITY=NETWORK          transmission priority
         LINEROW  WEIGHT=30,                line row weight            *
               NUMBER=1,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(4M,MAXIMUM),       line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=1,                 node row number            *
               WEIGHT=5,                    node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,31)              route addition resistance
         LINEROW  WEIGHT=60,                line row weight            *
               NUMBER=2,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(56000,MAXIMUM),    line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=2,                 node row number            *
               WEIGHT=10,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,63)              route addition resistance
         LINEROW  WEIGHT=90,                line row weight            *
               NUMBER=3,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(19200,MAXIMUM),    line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=3,                 node row number            *
               WEIGHT=20,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,95)              route addition resistance
         LINEROW  WEIGHT=120,               line row weight            *
               NUMBER=4,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(9600,MAXIMUM),     line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=4,                 node row number            *
               WEIGHT=40,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,127)             route addition resistance
         LINEROW  WEIGHT=150,               line row weight            *
               NUMBER=5,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(19200,MAXIMUM),    LINE SPEED                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,PACKET),     propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
```

```
          NODEROW  NUMBER=5,                 node row number            *
                WEIGHT=60,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,159)             route addition resistance
          LINEROW  WEIGHT=180,               line row weight            *
                NUMBER=6,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(9600,MAXIMUM),     LINE SPEED                 *
                COSTTIME=(0,128),            cost per connect time      *
                COSTBYTE=(0,128),            cost per byte transmitted  *
                PDELAY=(MINIMUM,PACKET),     propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=6,                 node row number            *
                WEIGHT=80,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,191)             route addition resistance
          LINEROW  WEIGHT=210,               line row weight            *
                NUMBER=7,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4800,MAXIMUM),     LINE SPEED                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=7,                 node row number            *
                WEIGHT=120,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,223)             route addition resistance
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=8,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#CONNECT APPNCOS  PRIORITY=MEDIUM            transmission priority
          LINEROW  WEIGHT=30,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4M,MAXIMUM),       line speed                 *
                COSTTIME=(0,0),              cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
```

```
      CAPACITY=(56000,MAXIMUM),    line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=2,                 node row number            *
      WEIGHT=10,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,63)              route addition resistance
LINEROW  WEIGHT=90,                line row weight            *
      NUMBER=3,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=3,                 node row number            *
      WEIGHT=20,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,95)              route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=4,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=150,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(19200,MAXIMUM),    LINE SPEED                 *
      COSTTIME=(0,0),              cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     LINE SPEED                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=210,               line row weight            *
      NUMBER=7,                    line row number            *
```

```
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(4800,MAXIMUM),     LINE SPEED                 *
                COSTTIME=(0,196),            cost per connect time      *
                COSTBYTE=(0,196),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=7,                 node row number            *
                WEIGHT=120,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,223)             route addition resistance
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=8,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
*
#INTER    APPNCOS  PRIORITY=HIGH,NUMBER=12   transmission priority
          LINEROW  WEIGHT=20,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(100M,MAXIMUM),     line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=40,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(10M,MAXIMUM),      line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=2,                 node row number            *
                WEIGHT=10,                   node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,63)              route addition resistance
          LINEROW  WEIGHT=60,                line row weight            *
                NUMBER=3,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(1M,MAXIMUM),       line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=3,                 node row number            *
```

```
      WEIGHT=20,                    node row weight           *
      CONGEST=(LOW,LOW),            congestion                *
      ROUTERES=(0,95)               route addition resistance
LINEROW  WEIGHT=80,                 line row weight           *
      NUMBER=4,                     line row number           *
      UPARM1=(0,255),               user defined char 1       *
      UPARM2=(0,255),               user defined char 2       *
      UPARM3=(0,255),               user defined char 3       *
      CAPACITY=(10M,MAXIMUM),       line speed                *
      COSTTIME=(0,64),              cost per connect time     *
      COSTBYTE=(0,0),               cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),    propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=4,                  node row number           *
      WEIGHT=40,                    node row weight           *
      CONGEST=(LOW,LOW),            congestion                *
      ROUTERES=(0,127)              route addition resistance
LINEROW  WEIGHT=100,                line row weight           *
      NUMBER=5,                     line row number           *
      UPARM1=(0,255),               user defined char 1       *
      UPARM2=(0,255),               user defined char 2       *
      UPARM3=(0,255),               user defined char 3       *
      CAPACITY=(1M,MAXIMUM),        line speed                *
      COSTTIME=(0,64),              cost per connect time     *
      COSTBYTE=(0,0),               cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),    propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=5,                  node row number           *
      WEIGHT=60,                    node row weight           *
      CONGEST=(LOW,LOW),            congestion                *
      ROUTERES=(0,159)              route addition resistance
LINEROW  WEIGHT=120,                line row weight           *
      NUMBER=6,                     line row number           *
      UPARM1=(0,255),               user defined char 1       *
      UPARM2=(0,255),               user defined char 2       *
      UPARM3=(0,255),               user defined char 3       *
      CAPACITY=(56000,MAXIMUM),     line speed                *
      COSTTIME=(0,64),              cost per connect time     *
      COSTBYTE=(0,0),               cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),    propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=6,                  node row number           *
      WEIGHT=80,                    node row weight           *
      CONGEST=(LOW,LOW),            congestion                *
      ROUTERES=(0,191)              route addition resistance
LINEROW  WEIGHT=140,                line row weight           *
      NUMBER=7,                     line row number           *
      UPARM1=(0,255),               user defined char 1       *
      UPARM2=(0,255),               user defined char 2       *
      UPARM3=(0,255),               user defined char 3       *
      CAPACITY=(1M,MAXIMUM),        line speed                *
      COSTTIME=(0,128),             cost per connect time     *
      COSTBYTE=(0,128),             cost per byte transmitted *
      PDELAY=(MINIMUM,TERRESTR),    propagation delay         *
      SECURITY=(UNSECURE,MAXIMUM)   security level for TG
NODEROW  NUMBER=7,                  node row number           *
      WEIGHT=120,                   node row weight           *
      CONGEST=(LOW,HIGH),           congestion                *
      ROUTERES=(0,223)              route addition resistance
LINEROW  WEIGHT=160,                line row weight           *
      NUMBER=8,                     line row number           *
      UPARM1=(0,255),               user defined char 1       *
      UPARM2=(0,255),               user defined char 2       *
      UPARM3=(0,255),               user defined char 3       *
      CAPACITY=(56000,MAXIMUM),     line speed                *
      COSTTIME=(0,128),             cost per connect time     *
      COSTBYTE=(0,128),             cost per byte transmitted *
```

```
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          NODEROW  NUMBER=8,                 node row number            *
                WEIGHT=160,                  node row weight            *
                CONGEST=(LOW,HIGH),          congestion                 *
                ROUTERES=(0,255)             route addition resistance
          LINEROW  WEIGHT=180,               line row weight            *
                NUMBER=9,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(29K,MAXIMUM),      line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          LINEROW  WEIGHT=200,               line row weight            *
                NUMBER=10,                   line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(29K,MAXIMUM),      line speed                 *
                COSTTIME=(0,128),            cost per connect time      *
                COSTBYTE=(0,128),            cost per byte transmitted  *
                PDELAY=(MINIMUM,PACKET),     propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          LINEROW  WEIGHT=220,               line row weight            *
                NUMBER=11,                   line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(9600,MAXIMUM),     line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,PACKET),     propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
          LINEROW  WEIGHT=240,               line row weight            *
                NUMBER=12,                   line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
                COSTTIME=(0,255),            cost per connect time      *
                COSTBYTE=(0,255),            cost per byte transmitted  *
                PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
                SECURITY=(UNSECURE,MAXIMUM)  security level for TG
*
#INTERSC APPNCOS  PRIORITY=HIGH,NUMBER=12    transmission priority
          LINEROW  WEIGHT=20,                line row weight            *
                NUMBER=1,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
                CAPACITY=(100M,MAXIMUM),     line speed                 *
                COSTTIME=(0,64),             cost per connect time      *
                COSTBYTE=(0,0),              cost per byte transmitted  *
                PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
                SECURITY=(PUBLIC,MAXIMUM)    security level for TG
          NODEROW  NUMBER=1,                 node row number            *
                WEIGHT=5,                    node row weight            *
                CONGEST=(LOW,LOW),           congestion                 *
                ROUTERES=(0,31)              route addition resistance
          LINEROW  WEIGHT=40,                line row weight            *
                NUMBER=2,                    line row number            *
                UPARM1=(0,255),              user defined char 1        *
                UPARM2=(0,255),              user defined char 2        *
                UPARM3=(0,255),              user defined char 3        *
```

```
      CAPACITY=(10M,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=2,                 node row number            *
      WEIGHT=10,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,63)              route addition resistance
LINEROW  WEIGHT=60,                line row weight            *
      NUMBER=3,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(1M,MAXIMUM),       line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,NEGLIGIB),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=3,                 node row number            *
      WEIGHT=20,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,95)              route addition resistance
LINEROW  WEIGHT=80,                line row weight            *
      NUMBER=4,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(10M,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=100,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(1M,MAXIMUM),       line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(56000,MAXIMUM),    line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=140,               line row weight            *
      NUMBER=7,                    line row number            *
```

```
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(1M,MAXIMUM),       line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=7,                 node row number            *
      WEIGHT=120,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,223)             route addition resistance
LINEROW  WEIGHT=160,               line row weight            *
      NUMBER=8,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(56000,MAXIMUM),    line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=8,                 node row number            *
      WEIGHT=160,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,255)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=9,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(29K,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,TERRESTR),   propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
LINEROW  WEIGHT=200,               line row weight            *
      NUMBER=10,                   line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(29K,MAXIMUM),      line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
LINEROW  WEIGHT=220,               line row weight            *
      NUMBER=11,                   line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(9600,MAXIMUM),     line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,PACKET),     propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
LINEROW  WEIGHT=240,               line row weight            *
      NUMBER=12,                   line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(MINIMUM,MAXIMUM),  line speed                 *
      COSTTIME=(0,255),            cost per connect time      *
      COSTBYTE=(0,255),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
*
```

APPN COS Definitions

```
#BATCH   APPNCOS  PRIORITY=LOW,NUMBER=12    transmission priority
         LINEROW  WEIGHT=20,                line row weight            *
               NUMBER=1,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(100M,MAXIMUM),     line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=1,                 node row number            *
               WEIGHT=5,                    node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,31)              route addition resistance
         LINEROW  WEIGHT=40,                line row weight            *
               NUMBER=2,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(10M,MAXIMUM),      line speed                 *
               COSTTIME=(0,0),              cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=2,                 node row number            *
               WEIGHT=10,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,63)              route addition resistance
         LINEROW  WEIGHT=60,                line row weight            *
               NUMBER=3,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(10M,MAXIMUM),      line speed                 *
               COSTTIME=(0,64),             cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=3,                 node row number            *
               WEIGHT=20,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,95)              route addition resistance
         LINEROW  WEIGHT=80,                line row weight            *
               NUMBER=4,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(1M,MAXIMUM),       line speed                 *
               COSTTIME=(0,64),             cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
         NODEROW  NUMBER=4,                 node row number            *
               WEIGHT=40,                   node row weight            *
               CONGEST=(LOW,LOW),           congestion                 *
               ROUTERES=(0,127)             route addition resistance
         LINEROW  WEIGHT=100,               line row weight            *
               NUMBER=5,                    line row number            *
               UPARM1=(0,255),              user defined char 1        *
               UPARM2=(0,255),              user defined char 2        *
               UPARM3=(0,255),              user defined char 3        *
               CAPACITY=(56000,MAXIMUM),    line speed                 *
               COSTTIME=(0,64),             cost per connect time      *
               COSTBYTE=(0,0),              cost per byte transmitted  *
               PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
               SECURITY=(UNSECURE,MAXIMUM)  security level for TG
```

```
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(29K,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=140,               line row weight            *
      NUMBER=7,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(100M,MAXIMUM),     line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=7,                 node row number            *
      WEIGHT=120,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,223)             route addition resistance
LINEROW  WEIGHT=160,               line row weight            *
      NUMBER=8,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(10M,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
NODEROW  NUMBER=8,                 node row number            *
      WEIGHT=160,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,255)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
      NUMBER=9,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(56000,MAXIMUM),    line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
LINEROW  WEIGHT=200,               line row weight            *
      NUMBER=10,                   line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(29K,MAXIMUM),      line speed                 *
      COSTTIME=(0,128),            cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(UNSECURE,MAXIMUM)  security level for TG
LINEROW  WEIGHT=220,               line row weight            *
```

```
                NUMBER=11,                  line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(9600,MAXIMUM),    line speed                *
                COSTTIME=(0,64),            cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
          LINEROW  WEIGHT=240,              line row weight           *
                NUMBER=12,                  line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(MINIMUM,MAXIMUM), line speed                *
                COSTTIME=(0,255),           cost per connect time     *
                COSTBYTE=(0,255),           cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(UNSECURE,MAXIMUM) security level for TG
*
#BATCHSC APPNCOS  PRIORITY=LOW,NUMBER=12    transmission priority
          LINEROW  WEIGHT=20,               line row weight           *
                NUMBER=1,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(100M,MAXIMUM),    line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=1,                node row number           *
                WEIGHT=5,                   node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,31)             route addition resistance
          LINEROW  WEIGHT=40,               line row weight           *
                NUMBER=2,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(10M,MAXIMUM),     line speed                *
                COSTTIME=(0,0),             cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=2,                node row number           *
                WEIGHT=10,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,63)             route addition resistance
          LINEROW  WEIGHT=60,               line row weight           *
                NUMBER=3,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
                UPARM3=(0,255),             user defined char 3       *
                CAPACITY=(10M,MAXIMUM),     line speed                *
                COSTTIME=(0,64),            cost per connect time     *
                COSTBYTE=(0,0),             cost per byte transmitted *
                PDELAY=(MINIMUM,MAXIMUM),   propagation delay         *
                SECURITY=(PUBLIC,MAXIMUM)   security level for TG
          NODEROW  NUMBER=3,                node row number           *
                WEIGHT=20,                  node row weight           *
                CONGEST=(LOW,LOW),          congestion                *
                ROUTERES=(0,95)             route addition resistance
          LINEROW  WEIGHT=80,               line row weight           *
                NUMBER=4,                   line row number           *
                UPARM1=(0,255),             user defined char 1       *
                UPARM2=(0,255),             user defined char 2       *
```

```
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(1M,MAXIMUM),       line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=4,                 node row number            *
      WEIGHT=40,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,127)             route addition resistance
LINEROW  WEIGHT=100,               line row weight            *
      NUMBER=5,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(56000,MAXIMUM),    line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=5,                 node row number            *
      WEIGHT=60,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,159)             route addition resistance
LINEROW  WEIGHT=120,               line row weight            *
      NUMBER=6,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(29K,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,0),              cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=6,                 node row number            *
      WEIGHT=80,                   node row weight            *
      CONGEST=(LOW,LOW),           congestion                 *
      ROUTERES=(0,191)             route addition resistance
LINEROW  WEIGHT=140,               line row weight            *
      NUMBER=7,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(100M,MAXIMUM),     line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=7,                 node row number            *
      WEIGHT=120,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,223)             route addition resistance
LINEROW  WEIGHT=160,               line row weight            *
      NUMBER=8,                    line row number            *
      UPARM1=(0,255),              user defined char 1        *
      UPARM2=(0,255),              user defined char 2        *
      UPARM3=(0,255),              user defined char 3        *
      CAPACITY=(10M,MAXIMUM),      line speed                 *
      COSTTIME=(0,64),             cost per connect time      *
      COSTBYTE=(0,128),            cost per byte transmitted  *
      PDELAY=(MINIMUM,MAXIMUM),    propagation delay          *
      SECURITY=(PUBLIC,MAXIMUM)    security level for TG
NODEROW  NUMBER=8,                 node row number            *
      WEIGHT=160,                  node row weight            *
      CONGEST=(LOW,HIGH),          congestion                 *
      ROUTERES=(0,255)             route addition resistance
LINEROW  WEIGHT=180,               line row weight            *
```

```
                    NUMBER=9,                    line row number           *
                    UPARM1=(0,255),              user defined char 1       *
                    UPARM2=(0,255),              user defined char 2       *
                    UPARM3=(0,255),              user defined char 3       *
                    CAPACITY=(56000,MAXIMUM),    line speed                *
                    COSTTIME=(0,128),            cost per connect time     *
                    COSTBYTE=(0,128),            cost per byte transmitted *
                    PDELAY=(MINIMUM,MAXIMUM),    propagation delay         *
                    SECURITY=(PUBLIC,MAXIMUM)    security level for TG
              LINEROW  WEIGHT=200,               line row weight           *
                    NUMBER=10,                   line row number           *
                    UPARM1=(0,255),              user defined char 1       *
                    UPARM2=(0,255),              user defined char 2       *
                    UPARM3=(0,255),              user defined char 3       *
                    CAPACITY=(29K,MAXIMUM),      line speed                *
                    COSTTIME=(0,128),            cost per connect time     *
                    COSTBYTE=(0,128),            cost per byte transmitted *
                    PDELAY=(MINIMUM,MAXIMUM),    propagation delay         *
                    SECURITY=(PUBLIC,MAXIMUM)    security level for TG
              LINEROW  WEIGHT=220,               line row weight           *
                    NUMBER=11,                   line row number           *
                    UPARM1=(0,255),              user defined char 1       *
                    UPARM2=(0,255),              user defined char 2       *
                    UPARM3=(0,255),              user defined char 3       *
                    CAPACITY=(9600,MAXIMUM),     line speed                *
                    COSTTIME=(0,64),             cost per connect time     *
                    COSTBYTE=(0,0),              cost per byte transmitted *
                    PDELAY=(MINIMUM,MAXIMUM),    propagation delay         *
                    SECURITY=(PUBLIC,MAXIMUM)    security level for TG
              LINEROW  WEIGHT=240,               line row weight           *
                    NUMBER=12,                   line row number           *
                    UPARM1=(0,255),              user defined char 1       *
                    UPARM2=(0,255),              user defined char 2       *
                    UPARM3=(0,255),              user defined char 3       *
                    CAPACITY=(MINIMUM,MAXIMUM),  line speed                *
                    COSTTIME=(0,255),            cost per connect time     *
                    COSTBYTE=(0,255),            cost per byte transmitted *
                    PDELAY=(MINIMUM,MAXIMUM),    propagation delay         *
                    SECURITY=(PUBLIC,MAXIMUM)    security level for TG
*
```

APPN Transmission Group (TG) Profile Definitions (IBMTGPS)

Ethernet

```
ETHERNET TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=10M
```

Token Ring

```
TOKNRING TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=4M
```

ISDN Non-Switched

```
ISDNNSWT TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=64K
```

ISDN Switched

```
ISDNSWTD TGP   COSTTIME=128,COSTBYTE=128,SECURITY=UNSECURE,            *
               PDELAY=TERRESTR,CAPACITY=10K
```

SDLC Non-Switched

```
SDLCNSWT TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=10K
```

SDLC Switched

```
SDLCSWTD TGP   COSTTIME=128,COSTBYTE=128,SECURITY=UNSECURE,            *
               PDELAY=TERRESTR,CAPACITY=10K
```

X.25

```
X25      TGP   COSTTIME=128,COSTBYTE=128,SECURITY=PUBLIC,              *
               PDELAY=PACKET,CAPACITY=10K
```

16M Token Ring

```
TRING16M TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=16M
```

Channel

```
CHANNEL  TGP   COSTTIME=0,COSTBYTE=0,SECURITY=SECURE,                  *
               PDELAY=NEGLIGIB,CAPACITY=36M
```

ESCON Channel

```
ESCON    TGP   COSTTIME=0,COSTBYTE=0,SECURITY=SECURE,                  *
               PDELAY=NEGLIGIB,CAPACITY=200M
```

ATM TGs over PVCs (Campus [Private] ATM Best Effort)

```
ATMPVCCB TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=25M
```

ATM TGs over PVCs (Campus ATM Reserved Bandwidth)

```
ATMPVCCR TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=NEGLIGIB,CAPACITY=25M
```

ATM TGs over PVCs (WAN [Public] ATM Best Effort)

```
ATMPVCWB TGP   COSTTIME=0,COSTBYTE=128,SECURITY=PUBLIC,                *
               PDELAY=PACKET,CAPACITY=25M
```

ATM TGs over PVCs (WAN ATM Reserved Bandwidth)

```
ATMPVCWR TGP   COSTTIME=0,COSTBYTE=0,SECURITY=PUBLIC,                  *
               PDELAY=TERRESTR,CAPACITY=25M
```

ATM TGs over SVCs (Campus ATM Best Effort)

```
ATMSVCCB TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=25M
```

ATM TGs over SVCs (Campus ATM Reserved Bandwidth)

```
ATMSVCCR TGP   COSTTIME=64,COSTBYTE=0,SECURITY=UNSECURE,               *
               PDELAY=NEGLIGIB,CAPACITY=25M
```

ATM TGs over SVCs (WAN ATM Best Effort)

```
ATMSVCWB TGP   COSTTIME=0,COSTBYTE=128,SECURITY=PUBLIC,                *
               PDELAY=PACKET,CAPACITY=25M
```

ATM TGs over SVCs (WAN ATM Reserved Bandwidth)

```
ATMSVCWR TGP   COSTTIME=128,COSTBYTE=0,SECURITY=PUBLIC,                *
               PDELAY=TERRESTR,CAPACITY=25M
```

Enterprise Extender, Campus

```
EEXTCAMP TGP   COSTTIME=0,COSTBYTE=0,SECURITY=UNSECURE,                *
               PDELAY=TERRESTR,CAPACITY=4M
```

Enterprise Extender, Wide Area Network

```
EEXTWAN  TGP   COSTTIME=0,COSTBYTE=0,SECURITY=PUBLIC,                  *
               PDELAY=PACKET,CAPACITY=56K
```

Message Flooding Table (ISTMSFLD)

```
ISTMSFLD FLDTAB
*/*******************************************************
*/*   IST017I                                           *
*/*      UNABLE TO LOAD PHASE %                         *
*/*******************************************************
M017     FLDENT  MESSAGE=0017,                  MESSAGE IST017I        X
               TIME=,                           DEFAULT TO 30 SECONDS  X
               VARTEXT=(YES),                   PHASE NAME             X
               HARDCOPY=                        DEFAULT TO YES
*/*******************************************************
*/*   IST118I                                           *
*/*      ANOMALY FOUND NEAR RECORD % IN MEMBER % - CODE % *
*/*******************************************************
M118     FLDENT  MESSAGE=0118,                  MESSAGE IST118I        X
               TIME=,                           DEFAULT TO 30 SECONDS  X
               VARTEXT=(NO,                     COUNT                  X
               YES,                             MEMBER NAME            X
               YES),                            CODE                   X
               HARDCOPY=                        DEFAULT TO YES
*/*******************************************************
*/*   IST154I                                           *
*/*      EXPANSION FAILED FOR % BUFFER POOL - CODE %,     *
*/*      USERID = %                                     *
*/*******************************************************
M154     FLDENT  MESSAGE=0154,                  MESSAGE IST154I        X
               TIME=,                           DEFAULT TO 30 SECONDS  X
               VARTEXT=(YES,                    BUFFER POOL            X
               YES,                             CODE                   X
               YES),                            USERID                 X
               HARDCOPY=                        DEFAULT TO YES
*/*******************************************************
*/*   IST182I                                           *
*/*      UNABLE TO GET STORAGE FOR CKPT % MAJOR NODE %    *
*/*******************************************************
M182     FLDENT  MESSAGE=0182,                 MESSAGE IST182I         X
               TIME=,                           DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                    DATASET NAME            X
               YES),                            NODE NAME               X
               HARDCOPY=                        DEFAULT TO YES
*/*******************************************************
*/*   IST192I                                           *
*/*      POA MSG TRANSFER FAILED - INSUFFICIENT STORAGE   *
*/*******************************************************
M192     FLDENT  MESSAGE=0192,                 MESSAGE IST192I         X
               TIME=,                           DEFAULT TO 30 SECONDS   X
               HARDCOPY=                        DEFAULT TO YES
*/*******************************************************
*/*   IST208I                                           *
*/*      UNABLE TO TERMINATE SESSIONS FOR ID = % -        *
```

```
*/*      INSUFFICIENT STORAGE                            *
*/*********************************************************
M208     FLDENT  MESSAGE=0208,                 MESSAGE IST208I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=YES,                    MINOR NODE              X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST234I                                            *
*/*      I/O ERROR %                                     *
*/*********************************************************
M234     FLDENT  MESSAGE=0234,                 MESSAGE IST234I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=YES,                    ONE TEXT FIELD          X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST301I                                            *
*/*      INSUFFICIENT VFBUF STORAGE FOR I/O TRACE TABLE   *
*/*********************************************************
M301     FLDENT  MESSAGE=0301,                 MESSAGE IST301I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST348I                                            *
*/*      UNABLE TO PROCESS DISCONNECTION FOR PU = %       *
*/*      DUE TO LACK OF STORAGE                          *
*/*********************************************************
M348     FLDENT  MESSAGE=0348,                 MESSAGE IST348I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                  PUNAME                  X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST366I                                            *
*/*      CONFIG % UNABLE TO DEFINE % - MAXIMUM NUMBER OF  *
*/*      NETWORK ADDRESSES FOR HOST SUBAREA EXCEEDED      *
*/*********************************************************
M366     FLDENT  MESSAGE=0366,                 MESSAGE IST366I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                   CONFIG NAME             X
               NO),                            NODE NAME               X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST367I                                            *
*/*      NO STORAGE TO DEFINE NODE % CONFIG %             *
*/*********************************************************
M367     FLDENT  MESSAGE=0367,                 MESSAGE IST367I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(NO,                    NODE NAME               X
               YES),                           CONFIG NAME             X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST380I                                            *
*/*      ERROR FOR ID = % - REQUEST: %, SENSE %           *
*/*********************************************************
M380     FLDENT  MESSAGE=0380,                 MESSAGE IST380I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(NO,                    NODE NAME               X
               YES,                            RU NAME                 X
               YES),                           SENSE CODE              X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST436I                                            *
*/*      STORAGE NOT AVAILABE FOR TUNING STATISTICS DATA  *
*/*********************************************************
M436     FLDENT  MESSAGE=0436,                 MESSAGE IST436I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
```

```
*/*   IST522I                                            *
*/*      % % % SA % TO SA % %                            *
*/*******************************************************
M522     FLDENT  MESSAGE=0522,                MESSAGE IST522I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                  ER/VR NUMBER            X
               YES,                           OPTION                  X
               YES,                           FAILED/REJECTED         X
               YES,                           SA1                     X
               YES,                           SA2                     X
               YES),                          FOR TPi                 X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST526I                                            *
*/*      ROUTE FAILED FROM % TO % - DSA % - NETID %      *
*/*******************************************************
M526     FLDENT  MESSAGE=0526,                MESSAGE IST526I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                  SUBAREA1                X
               YES,                           SUBAREA2                X
               YES,                           DESTSUBAREA             X
               YES),                          NETID                   X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST561I                                            *
*/*      STORAGE UNAVAILABLE: % BUFFER POOL              *
*/*******************************************************
M561     FLDENT  MESSAGE=0561,                MESSAGE IST561I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                 BUFFER POOL             X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST562I                                            *
*/*      STORAGE UNAVAILABLE: % REACHED                  *
*/*******************************************************
M562     FLDENT  MESSAGE=0562,                MESSAGE IST562I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                 CSA TYPE                X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST563I                                            *
*/*      STORAGE UNAVAILABLE: MAXPVT REACHED FOR % %     *
*/*******************************************************
M563     FLDENT  MESSAGE=0563,                MESSAGE IST563I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                  JOB NAME                X
               YES),                          STEPNAME|USERID         X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST564I                                            *
*/*      STORAGE UNAVAILABLE: COMMON AREA SUBPOOL %      *
*/*******************************************************
M564     FLDENT  MESSAGE=0564,                MESSAGE IST564I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                 SUBPOOL                 X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST565I                                            *
*/*      STORAGE UNAVAILABLE: VTAM PRIVATE AREA SUBPOOL % *
*/*******************************************************
M565     FLDENT  MESSAGE=0565,                MESSAGE IST565I         X
               TIME=,                         DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                 SUBPOOL                 X
               HARDCOPY=                      DEFAULT TO YES
*/*******************************************************
*/*   IST566I                                            *
*/*      STORAGE UNAVAILABLE: % % SUBPOOL %              *
```

```
*/*********************************************************
M566     FLDENT  MESSAGE=0566,                 MESSAGE IST566I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                   JOB NAME                X
               YES,                            STEPNAME|USERID         X
               YES),                           SUBPOOL NAME            X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST663I                                             *
*/*      % REQUEST %, SENSE = %                           *
*/*********************************************************
M663     FLDENT  MESSAGE=0663,                 MESSAGE IST663I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                   REQUEST                 X
               YES,                            ADJNODE NAME - RESULTS  X
               YES),                           SENSE                   X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST678I                                             *
*/*      INSUFFICIENT STORAGE TO SCHEDULE TPEND EXIT      *
*/*      FOR %                                            *
*/*********************************************************
M678     FLDENT  MESSAGE=0678,                 MESSAGE IST678I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                  APPL NAME               X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST820I                                             *
*/*      % RSP DATA DISCARDED FOR ID = % - INSUFF STORAGE *
*/*********************************************************
M820     FLDENT  MESSAGE=0820,                 MESSAGE IST820I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                   'ACTLU'|'ACTPU'         X
               YES),                           NODE NAME               X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST822I                                             *
*/*      CRRM % RECOVERY FAILED - INSUFFICIENT STORAGE    *
*/*********************************************************
M822     FLDENT  MESSAGE=0822,                 MESSAGE IST822I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                  CDRM NAME               X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST860I                                             *
*/*      DEACTIVATION  OF % INCOMPLETE  - INSUFFICIENT    *
*/*      STORAGE                                          *
*/*********************************************************
M860     FLDENT  MESSAGE=0860,                 MESSAGE IST860I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                  NODE NAME               X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST930I                                             *
*/*      % - % SESSION USING % OF %BUF                    *
*/*********************************************************
M930     FLDENT  MESSAGE=0930,                 MESSAGE IST930I         X
               TIME=,                          DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                   NODENAME1               X
               YES,                            NODENAME2               X
               YES,                            PERCENTAGE              X
               YES),                           POOL                    X
               HARDCOPY=                       DEFAULT TO YES
*/*********************************************************
*/*   IST932E                                             *
*/*      FAILURE OCCURRED DURING TAKEOVER OF %, SENSE=%   *
*/*********************************************************
```

```
M932     FLDENT  MESSAGE=0932,                MESSAGE IST932E         X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                 LUNAME                  X
               YES),                         SENSE CODE              X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST970I                                            *
*/*      LU - LU VERIFICATION ERROR % FOR %              *
*/*********************************************************
M970     FLDENT  MESSAGE=0970,                MESSAGE IST970I         X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                 CODE                    X
               YES),                         PROFILE NAME            X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST976I                                            *
*/*      ENTRY % DEFINED BUT NO % DEFINED FOR %          *
*/*********************************************************
M976     FLDENT  MESSAGE=0976,                MESSAGE IST976I         X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                 ENTRY NAME              X
               NO,                           TABLE TYPE              X
               NO),                          RESOURCE NAME           X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST983E                                            *
*/*      % MESSAGE QUEUE LIMIT EXCEEDED - FURTHER        *
*/*      MESSAGES WILL BE DISCARDED                      *
*/*********************************************************
M983     FLDENT  MESSAGE=0983,                MESSAGE IST983E         X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                POA NAME                X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST999E                                            *
*/*      VTAM MESSAGE LOST - INSUFFICIENT STORAGE        *
*/*********************************************************
M999     FLDENT  MESSAGE=0999,                MESSAGE IST999E         X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST1000I                                           *
*/*      STORAGE UNAVAILABLE IN %                        *
*/*********************************************************
M1000    FLDENT  MESSAGE=1000,                MESSAGE IST1000I        X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES),                REASON CODE             X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST1011I                                           *
*/*      ENTRY % NOT FOUND IN % % FOR %                  *
*/*********************************************************
M1011    FLDENT  MESSAGE=1011,                MESSAGE IST1011I        X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                 ENTRY NAME              X
               NO,                           TABLE TYPE              X
               YES,                          TABLE NAME              X
               NO),                          RESOURCE NAME           X
               HARDCOPY=                     DEFAULT TO YES
*/*********************************************************
*/*   IST1139I
*/*      % FOR % FAILED - SENSE: %                       *
*/*********************************************************
M1139    FLDENT  MESSAGE=1139,                MESSAGE IST1139I        X
               TIME=,                        DEFAULT TO 30 SECONDS   X
               VARTEXT=(YES,                 RU                      X
               NO,                           NODE NAME               X
```

```
              YES),                          SENSE                  X
              HARDCOPY=                      DEFAULT TO YES
*/*********************************************************
*/*   IST1153I
*/*      % % SESSION %BUF USE %                          *
*/*********************************************************
M1153    FLDENT  MESSAGE=1153,                MESSAGE IST1153I        X
              TIME=,                         DEFAULT TO 30 SECONDS   X
              VARTEXT=(YES,                  LU1                     X
              YES,                           LU2                     X
              YES,                           POOL                    X
              YES),                          PERCENT                 X
              HARDCOPY=                      DEFAULT TO YES
*/*********************************************************
*/*   IST1211I
*/*      I/O ERROR % % % %                               *
*/*********************************************************
M1211    FLDENT  MESSAGE=1211,                MESSAGE IST1211I        X
              TIME=,                         DEFAULT TO 30 SECONDS   X
              VARTEXT=(YES,                  NODE NAME               X
              YES,                           CMD MODIFIER            X
              YES,                           RESPONSE                X
              YES),                          SENSE                   X
              HARDCOPY=                      DEFAULT TO YES
*/*********************************************************
*/*   IST1213I
*/*      % LU-LU VERIFY ERROR %                          *
*/*********************************************************
M1213    FLDENT  MESSAGE=1213,                MESSAGE IST1213I        X
              TIME=,                         DEFAULT TO 30 SECONDS   X
              VARTEXT=(YES,                  PROFILE NAME            X
              YES),                          REASON CODE             X
              HARDCOPY=                      DEFAULT TO YES
*/*********************************************************
*/*   IST1386I
*/*  DYNAMIC ALLOCATION FAILED FOR % CODE = % REASON = %  *
*/*********************************************************
M1386    FLDENT  MESSAGE=1386,                MESSAGE IST1386I        X
              TIME=,                         DEFAULT TO 30 SECONDS   X
              VARTEXT=(YES,                  DEVICE                  X
              YES,                           RETURN CODE             X
              YES),                          REASON CODE             X
              HARDCOPY=                      DEFAULT TO YES
*/*********************************************************
*/*   IST1388I                                            *
*/*  SOCKET % CALL FAILED, TCP ERROR NUMBER = %           *
*/*********************************************************
M1388    FLDENT  MESSAGE=1388,                MESSAGE IST1388I        X
              TIME=,                         DEFAULT TO 30 SECONDS   X
              VARTEXT=(YES,                  SOCKET CALL NAME        X
              YES),                          SOCKET ERROR NUMBER     X
              HARDCOPY=                      DEFAULT TO YES
ISTMSFLD FLDEND
```

CMIP Services Directory Definition File (ACYDDF)

An IBM-supplied CMIP services directory definition file with specific code might have security definitions that can contradict your security definitions. Hence, IBM supplies only the following in the file:

```
# This is the VTAM OSI CMIP Directory Definition File
```

A more detailed example of a directory definition file is in the *OS/390 eNetwork Communications Server: SNA Resource Definition Samples*. For more information

on the file and CMIP services, refer to the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide* and see “CMIP Services Directory Definition File” on page 681.

Appendix B. Examples of USS Command Conversion

This appendix contains examples of how to code USS tables to convert commands. Each example includes the following:

- The USS table used
- The format for entering the command
- The converted command

Example 1: Parameters omitted

The VTAM operator command

```
F NET,TNSTAT
```

is issued. All parameters are omitted and, therefore, defaults are used during conversion. The following operation-level USS table is used:

```
USSTAB
USSCMD CMD=TNSTAT,FORMAT=BAL
USSPARM PARM=P1,REP=CNSL,DEFAULT=NO
USSPARM PARM=P2,REP=TIME,DEFAULT=60
USSEND
```

The converted command is

```
TNSTAT CNSL(NO) TIME(60)
```

Example 2: Positional and keyword parameters (FORMAT=PL1)

The terminal operator command

```
LOFF(PROG) T(COND)
```

is issued. Positional and keyword parameters are used with FORMAT=PL1. The following session-level USS table is used:

```
USSTAB
USSCMD CMD=LOFF,REP=LOGOFF,FORMAT=PL1
USSPARM PARM=P1,REP=APPLID
USSPARM PARM=T,REP=TYPE,DEFAULT=COND
USSEND
```

The converted command is

```
LOGOFF APPLID(PROG) TYPE(COND)
```

Example 3: Positional and keyword parameters (FORMAT=PL1)

The terminal operator command

```
LOFF (PGM) T
```

is issued. A null value is taken instead of a default value. Positional and keyword parameters are used with FORMAT=PL1. The following session-level USS table is used:

```
USSTAB
USSCMD CMD=LOFF,REP=LOGOFF,FORMAT=PL1
USSPARM PARM=P1,REP=APPLID
USSPARM PARM=T,REP=TYPE,DEFAULT=COND
USSEND
```

The converted command is

© Copyright IBM Corp. 1984, 1999

```
LOGOFF APPLID(PGM) TYPE( )
```

Because T was coded, the default value specified in the table is not used. If T had not been coded, TYPE(COND) would have resulted. TYPE( ) causes unconditional termination. Note that if you replace DEFAULT=COND with VALUE=COND in the table entry (USSPARM PARM=T,REP=TYPE,VALUE=COND), this command results in a TYPE(COND) instead of TYPE( ).

Example 4: Positional and keyword parameters (FORMAT=BAL)

The terminal operator command

```
LON PROGRAM,IN='7,3,JOhn'
```

is issued. Positional and keyword parameters are used with FORMAT=BAL.

The following session-level USS table is used:

```
USSTAB
USSCMD CMD=LON,REP=LOGON,FORMAT=BAL
USSPARM PARM=P1,REP=APPLID,DEFAULT=TESTPROG
USSPARM PARM=MODE,REP=LOGMODE,DEFAULT='PROMPT'
USSPARM PARM=IN,REP=DATA
USSEND
```

The converted command is

```
LOGON APPLID(PROGRAM) LOGMODE(PROMPT) DATA('7,3,JOhn')
```

Note that no character translation was performed on JOhn because there were single quotation marks around it. Note also that the single quotation marks around PROMPT in the default declaration have been deleted (by the assembler when the USS table was assembled).

Example 5: Using parameter not defined in USS table

The VTAM operator command

```
F NET,IMR,ID=PU12,OPT=ACT,RECLIM=200
```

is issued. A parameter (ID) is used that is not defined on the USS table.

The following operation-level USS table is used:

```
USSTAB
USSCMD CMD=IMR,FORMAT=BAL
USSPARM PARM=OPT,DEFAULT=ACT
USSPARM PARM=RECLIM,DEFAULT=10
USSEND
```

The converted command is

```
IMR ID(PU12) OPT(ACT) RECLIM(200)
```

Because ID is not defined by a USSPARM macroinstruction in the USS table, the keyword is not changed during conversion.

Example 6: Renaming keyword replaced by verb

The VTAM operator command

```
F NET,XYZ,ID=PU12
```

is issued. This example shows how to rename a keyword that is replaced by a verb.

The following VTAM operator USS table is used:

```
   USSTAB
*
   USSCMD CMD=MODIFY,FORMAT=BAL
   USSPARM PARM=XYZ,REP=VERB,VALUE=XYZ
*
   USSCMD CMD=XYZ,REP=IMR,FORMAT=BAL
   USSPARM PARM=OPT,DEFAULT=ACT
   USSPARM PARM=RECLIM,DEFAULT=10
   USSEND
```

This results, after the first pass, in the following command of the form

```
VERB(XYZ) ID(PU12)
```

which in turn, after the second pass, results in the following converted command

```
IMR ID(PU12) OPT(ACT) RECLIM(10)
```

Note that the FORMAT=BAL specification on CMD=XYZ is ignored. After the first pass, VTAM uses the PL/I format for command processing.

Example 7: Using interpret and USS tables

If you want to use the facilities of both the USS table and the interpret table, the SEQNCE parameters in the interpret table should be coded to avoid a match during the first search of the interpret table (which would prevent the USS table from being used). Instead, a USSPARM statement in the USS table should be used to specify an APPLID that will match an interpret table entry during the second pass.

- The interpret table is coded as follows:

```
XRFINTAB INTAB
         LOGCHAR  APPLID=(USERVAR,ZZIMSUV),SEQNCE='ZZIMS'
         ENDINTAB XRFINTAB
```

- The USS table is coded as follows:

```
XRFUSTAB USSTAB
         USSCMD   CMD=IMS,REP=LOGON,FORMAT=PL1
         USSPARM  PARM=P1,REP=APPLID,DEFAULT=ZZIMS
         USSPARM  PARM=P2,REP=LOGMODE,DEFAULT=ILOGMODE
         USSEND
         END      XRFUSTAB
```

- If the LOGON sequence entered by the terminal user is “IMS,”
  - The interpret table will be searched for a matching entry, but none will be found.
  - The message will be translated, using the information in the USS table, to read

    ```
    LOGON APPLID(ZZIMS) LOGMODE(ILOGMODE)
    ```

  - The interpret table will be searched again, this time using “ZZIMS,” the string specified as the APPLID in the reformatted LOGON message. Because “ZZIMS” does match an entry in the interpret table, the information in that entry will be used. If the current value of the USERVAR ZZIMSUV is “IMS1,” the session will be established to the IMS1 application.
- On the other hand, if the LOGON sequence entered by the terminal user is “ZZIMS,” a match will be found during the initial search of the interpret table; Therefore, the USS table will not be used (the LOGMODE operand specified in the USS table will not be added). The USERVAR specified in the interpret table entry will still be used.

Command Conversion Examples

For further information on converting USS commands, see the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

Appendix C. VTAMTOPO Filtering Option Reporting

Table 94 on page 837 summarizes the results of using the VTAMTOPO filtering option for reporting a switched PU under an NCP.

The following is the legend for the table:

notIGNR/INCL
Neither IGNORE or INCLUDE specified.

r VTAMTOPO= not specified, REPORT inherited from node above.

R VTAMTOPO=REPORT specified (or NOLLINES at containing GROUP).

nr VTAMTOPO= not specified, NOREPORT inherited from node above.

NR VTAMTOPO=NOREPORT specified (or NOSWPUS at containing GROUP).

NotRep
Switched PU is not reported.

IGNR VTAMTOPO=IGNORE specified at designated major node.

INCL VTAMTOPO=INCLUDE specified at the designated major node.

any Value of VTAMTOPO= does not matter, inclusion not specified.

R-NCP
Switched PU is reported under the NCP under which the Switched PU is connected.

R-SSCP
Switched PU is reported under the SSCP directly (not under any NCP), as it is when it is not connected.

Notes:

1. Values shown are assumed to be set in the applicable major node, individual PUX or SW PU before the connection is established. A MODIFY VTAMTOPO to set these values after the connection is established may not show the expected result before the connection is taken down and reestablished.
2. This table can also be used for the Switched PUs that are connected under an XCA. In the results column, all R-NCP will be R-SSCP, since the XCA is not represented by a PU type, and does not appear as a discrete node in SNA local topology.
3. The PUX is a place holder for a future connected switched PU under the switched line. It takes its VTAMTOPO value either explicitly from the line GROUP value, or implicitly from the NCP or the XCA major node value. If its VTAMTOPO value is explicitly set, it cannot be modified.

Table 94. Connected Switched PU Report

VTAMTOPO value on				SPWU result
NCP	PUX	SWND	SWPU	
notIGNR/INCL	r/R	R	r	R-NCP
notIGNR/INCL	r/R	INCL	r	R-NCP
notIGNR/INCL	r/R	R	R	R-NCP
notIGNR/INCL	r/R	INCL	R	R-NCP
notIGNR/INCL	r/R	R	NR	NotRep
notIGNR/INCL	r/R	INCL	NR	R-NCP

© Copyright IBM Corp. 1984, 1999

Table 94. Connected Switched PU Report (continued)

VTAMTOPO value on				SPWU result
NCP	PUX	SWND	SWPU	
notIGNR/INCL	r/R	NR	nr	R-NCP
notIGNR/INCL	r/R	IGNR	nr	R-NCP
notIGNR/INCL	r/R	NR	NR	NotRep
notIGNR/INCL	r/R	IGNR	NR	NotRep
notIGNR/INCL	r/R	NR	R	R-NCP
notIGNR/INCL	r/R	IGNR	R	R-NCP
notIGNR/INCL	nr/NR	R	r	R-SSCP
notIGNR/INCL	nr/NR	INCL	r	R-SSCP
notIGNR/INCL	nr/NR	R	R	R-NCP
notIGNR/INCL	nr/NR	INCL	R	R-SSCP
notIGNR/INCL	nr/NR	R	NR	NotRep
notIGNR/INCL	nr/NR	INCL	NR	R-SSCP
notIGNR/INCL	nr/NR	NR	nr	NotRep
notIGNR/INCL	nr/NR	IGNR	nr	NotRep
notIGNR/INCL	nr/NR	NR	NR	NotRep
notIGNR/INCL	nr/NR	IGNR	NR	NotRep
notIGNR/INCL	nr/NR	NR	R	R-NCP
notIGNR/INCL	nr/NR	IGNR	R	NotRep
INCL	any	R	r	R-NCP
INCL	any	INCL	r	R-NCP
INCL	any	R	R	R-NCP
INCL	any	INCL	R	R-NCP
INCL	any	R	NR	NotRep
INCL	any	INCL	NR	R-NCP
INCL	any	NR	nr	R-NCP
INCL	any	IGNR	nr	R-NCP
INCL	any	NR	NR	NotRep
INCL	any	IGNR	NR	NotRep
INCL	any	NR	R	R-NCP
INCL	any	IGNR	R	R-NCP
IGNR	any	R	r	R-SSCP
IGNR	any	INCL	r	R-SSCP
IGNR	any	R	R	R-SSCP
IGNR	any	INCL	R	R-SSCP
IGNR	any	R	NR	NotRep
IGNR	any	INCL	NR	R-SSCP
IGNR	any	NR	nr	NotRep
IGNR	any	IGNR	nr	NotRep
IGNR	any	NotRep	NotRep	NotRep

Table 94. Connected Switched PU Report (continued)

VTAMTOPO value on				SPWU result
NCP	**PUX**	**SWND**	**SWPU**	
IGNR	any	IGNR	NR	NotRep
IGNR	any	NR	R	R-SSCP
IGNR	any	IGNR	R	NotRep

Appendix D. Information Apars

This appendix lists information apars for SNA-related books.

Notes:

1. Information apars contain updates to previous editions of the manuals listed below. Books updated for V2R8 contain all the updates except those contained in the information apars that may be issued after V2R8 books went to press.
2. Information apars are predefined for CS for OS/390 V2R8 and may not contain updates.

SNA Information Apars

Table 95 lists information apars for SNA-related books.

Table 95. SNA Information Apars

Title	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5	VTAM 4.4	VTAM 4.3	VTAM 4.2
Anynet SNA over TCP/IP (SC31-8578)	ii11922	ii11633	ii11624	ii11623			
Anynet Sockets over SNA (SC31-8577)	ii11921	ii11622	ii11519	ii11518	ii11494		
CSM Guide (SC31-8575)					ii10968		
IP and SNA Codes (SC31-8571)	ii11917	ii11611	ii11361	ii11097	ii10385		
SNA Customization (LY43-0110)	ii11923	ii11925	ii11924	ii11092 ii11621	ii10382	ii08792	
SNA Diagnosis (LY43-0079 and LY43-0080)	ii11915	ii11615	ii11357	ii11585	ii10387 ii10437	ii08786	
SNA Messages (SC31-8569)	ii11916	ii11610	ii11358	ii11096	ii10384	ii08787	
SNA Network Implementation Guide (SC31-8563)	ii11911	ii11609	ii11353 ii11493	ii11095	ii10383		
SNA Operation (SC31-8567)	ii11914	ii11612	ii11355	ii11098	ii10386	ii08541	ii08129
SNA Planning and Migration (SC31-8622)	ii11910	ii11614	ii11359	ii11100	ii10390		

© Copyright IBM Corp. 1984, 1999

Table 95. SNA Information Apars (continued)

Title	CS for OS/390 2.8	CS for OS/390 2.7	CS for OS/390 2.6	CS for OS/390 2.5	VTAM 4.4	VTAM 4.3	VTAM 4.2
SNA Programming (SC31-8573)	ii11920	ii11613	ii11360	ii11099	ii10389	ii09494	
SNA Quick Reference (SX75-0121)	ii11913						
SNA Resource Definition Reference (SC31-8565)	ii11912	ii11608	ii11354	ii11094 ii11151	ii10381 ii11463	ii08793	ii08126
SNA Resource Definition Samples (SC31-8566)					ii10509		

Appendix E. 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
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may 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.

© Copyright IBM Corp. 1984, 1999

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:

Site Counsel
IBM Corporation
P.O.Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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

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

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, photographs and color illustrations may not appear.

Some updates of this book, such as an update between OS/390 releases, might be available only in softcopy. You can obtain softcopy from the OS/390 Online Library Collection (SK2T-6700), the OS/390 PDF Library Collection (SK2T-6718), or the OS/390 Internet Library (http://www.ibm.com/s390/os390/). To order the latest hardcopy edition that is available, you might need to order a lower suffix (dash) level.

Programming Interface Information

This publication documents information NOT intended to be used as a Programming Interface of SecureWay Communications Server for OS/390.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AFP
AD/Cycle
AIX
AIX/ESA
AnyNet
APL2
APPN
AS/400
AT
BookManager
BookMaster
CBPDO
C/370
CICS
CICS/ESA
C/MVS
Common User Access
C Set ++
CT
CUA
DATABASE 2
DatagLANce
DB2
DFSMS
DFSMSdfp
DFSMShsm
DFSMS/MVS
Domino
DRDA
eNetwork
Enterprise Systems Architecture/370
ESA/390
ESCON
ES/3090
ES/9000
ES/9370
EtherStreamer
Extended Services
FAA
Micro Channel
MVS
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
MQ
Natural
NetView
Network Station
Nways
Notes
NTune
NTuneNCP
OfficeVision/MVS
OfficeVision/VM
Open Class
OpenEdition
OS/2
OS/390
Parallel Sysplex
Personal System/2
PR/SM
PROFS
PS/2
RACF
Resource Measurement Facility
RETAIN
RFM
RISC System/6000
RMF
RS/6000
S/370
S/390
SAA
SecureWay
Slate
SP
SP2
SQL/DS
System/360

FFST
FFST/2
FFST/MVS
First Failure Support Technology
GDDM
Hardware Configuration Definition
IBM
IBMLink
IMS
IMS/ESA
InfoPrint
Language Environment
LANStreamer
Library Reader
LPDA
MCS
System/370
System/390
SystemView
Tivoli
TURBOWAYS
UNIX System Services
Virtual Machine/Extended Architecture
VM/ESA
VM/XA
VSE/ESA
VTAM
WebSphere
XT
400
3090
3890

Lotus, Freelance, and Word Pro are trademarks of Lotus Development Corporation in the United States, or other countries, or both.

Tivoli and NetView are trademarks of Tivoli Systems Inc. in the United States, or other countries, or both.

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

UNIX is a registered trademark in the United States, other countries, or both and is licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ActionMedia, LANDesk, MMX, Pentium, and ProShare are trademarks of Intel Corporation in the United States, other countries, or both. For a complete list of Intel trademarks, see http://www.intel.com/tradmarx.htm.

Bibliography

SecureWay Communications Server for OS/390 Publications

Following are descriptions of the books in the SecureWay Communications Server for OS/390 library. The books are arranged in the following categories:

- Softcopy Information
- Planning
- Resource Definition, Configuration, and Tuning
- Operation
- Customization
- Writing Application Programs
- Diagnosis
- Messages and Codes
- APPC Application Suite.
- Multiprotocol Transport Networking (MPTN) Architecture publications

The complete set of unlicensed books in this section can be ordered using a single order number, SBOF-7011.

Updates to books are available on RETAIN. See “Appendix D. Information Apars” on page 841 for a list of the books and the INFOAPARS associated with them.

Some books are available in both hard- and soft-copy, or soft-copy only. The following abbreviations follow each order number:

HC/SC	Both hard- and soft-copy are available
SC	Only soft-copy is available

Related Publications

For information about OS/390 products, refer to *OS/390 Information Roadmap* (GC28-1727-03 [HC/SC]). The Roadmap describes what level of documents are supplied with each release of CS for OS/390, as well as describing each OS/390 publication.

Firewall

OS/390 Firewall Technologies Guide and Reference (SC24-5835-03 [HC/SC])

OSA-Express

S/390 Open Systems Adapter-Express Customer’s Guide and Reference (SA22-7403-01 [HC/SC])

Softcopy Information

- *OS/390 Online Library Collection* (SK2T-6700).

  This collection contains softcopy unlicensed books for OS/390, Parallel Sysplex products, and S/390 application programs that run on OS/390. This collection is updated quarterly with any new or updated books that are available for the product libraries included in it.
- *OS/390 PDF Library Collection* (SK2T-6718).

  This collection contains the unlicensed books for OS/390 Version 2 Release 6 in Portable Document Format (PDF).
- *OS/390 Licensed Product Library* (LK2T-2499).

  This library contains unencrypted softcopy licensed books for OS/390 Version 2. If any of the books in this library are changed, it is updated quarterly. The OS/390 Licensed Product Library for Version 1 (LK2T-6702) is still available, but is no longer updated.
- *System Center Publication IBM S/390 Redbooks Collection* (SK2T-2177).

  This collection contains over 300 ITSO redbooks that apply to the S/390 platform and to host networking arranged into subject bookshelves.

Planning

OS/390 SecureWay Communications Server: SNA Migration (SC31-8622-03 [HC/SC]). This book is intended to help you plan for SNA, whether you are migrating from a previous version or installing SNA for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with SNA.

OS/390 SecureWay Communications Server: IP Migration (SC31-8512-03 [HC/SC]). This book is intended to help you plan for IP, whether you are migrating from a previous version or installing IP for the first time. This book also identifies the optional and required modifications needed to enable you to use the enhanced functions provided with IP.

© Copyright IBM Corp. 1984, 1999

Resource Definition, Configuration, and Tuning

OS/390 SecureWay Communications Server: IP Configuration (SC31-8513-03 [HC/SC]). This book is for people who want to configure, customize, administer, and maintain IP. Familiarity with MVS operating system, IP protocols, and IBM Time Sharing Option (TSO) is recommended.

OS/390 SecureWay Communications Server: SNA Network Implementation Guide (SC31-8563-03 [HC/SC]). This book presents the major concepts involved in implementing a SNA network. Use this book in conjunction with the *OS/390 SecureWay Communications Server: SNA Resource Definition Reference*.

OS/390 SecureWay Communications Server: SNA Resource Definition Reference (SC31-8565-03 [HC/SC]). This book describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. The information includes:

- IBM-supplied default tables (logon mode and USS)
- Major node definitions
- User-defined tables and filters
- SNA start options.

Use this book in conjunction with the *OS/390 SecureWay Communications Server: SNA Network Implementation Guide*.

OS/390 eNetwork Communications Server: SNA Resource Definition Samples (SC31-8566-00 [HC/SC]). This book contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.

OS/390 eNetwork Communications Server: AnyNet SNA over TCP/IP (SC31-8578-00 [SC]). This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

OS/390 eNetwork Communications Server: AnyNet Sockets over SNA (SC31-8577-00 [SC]). This guide provides information to help you install, configure, use, and diagnose Sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.

Operation

OS/390 SecureWay Communications Server: IP User's Guide (GC31-8514-03 [HC/SC]). This book is for people who want to use TCP/IP for data communication activities such as FTP and Telnet. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

OS/390 SecureWay Communications Server: SNA Operation (SC31-8567-03 [HC/SC]). This book serves as a reference for programmers and operators requiring detailed information about specific operator commands.

OS/390 SecureWay Communications Server: Quick Reference (SX75-0121-03 [HC/SC]). This book contains essential information about SNA and IP operator commands.

OS/390 eNetwork Communications Server: High Speed Access Services User's Guide (GC31-8676-01 [SC]). This book is for end users and system administrators who want to use applications using a High Speed Access Services connection available in CS for OS/390.

Customization

OS/390 SecureWay Communications Server: SNA Customization (LY43-0110-01 [SC]). This book enables you to customize SNA, and includes:

- Communication network management (CNM) routing table
- Logon-interpret routine requirements
- Logon manager installation-wide exit routine for the CLU search exit
- TSO/SNA installation-wide exit routines
- SNA installation-wide exit routines

OS/390 eNetwork Communications Server: IP Network Print Facility (SC31-8522-00 [SC]). This book is for system programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP.

Writing Application Programs

OS/390 SecureWay Communications Server: IP Application Programming Interface Guide (SC31-8516-03 [SC]). This book describes the syntax and semantics of program source code necessary to write your own application

programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

OS/390 SecureWay Communications Server: IP CICS Sockets Guide (SC31-8518-01 [SC]). This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

OS/390 eNetwork Communications Server: IP IMS Sockets Guide (SC31-8519-00 [SC]). This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

OS/390 SecureWay Communications Server: IP Programmer's Reference (SC31-8515-03 [SC]). This book describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

OS/390 eNetwork Communications Server: SNA Programming (SC31-8573-01 [SC]). This book describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide (SC31-8581-00 [SC]). This book describes how to use the SNA LU 6.2 application programming interface for host application programs. This book applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this book.)

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference (SC31-8568-00 [SC]). This book provides reference material for the SNA LU 6.2 programming interface for host application programs.

OS/390 eNetwork Communications Server: CSM Guide (SC31-8575-00 [SC]). This book describes how applications use the communications storage manager.

OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide (SC31-8576-01 [SC]). This book describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The book provides guide and reference information about CMIP services and the SNA topology agent.

Diagnosis

OS/390 SecureWay Communications Server: IP Diagnosis (SC31-8521-03 [HC/SC]). This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

OS/390 SecureWay Communications Server: SNA Diagnosis Volume 1: Techniques and Procedures (LY43-0079-03 [HC/SC]) and *OS/390 SecureWay Communications Server: SNA Diagnosis Volume 2: FFST Dumps and the VIT* (LY43-0080-02 [HC/SC]). These books help you identify a SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation. Volume 1 includes information about the following:

- Command syntax for running traces and collecting and analyzing dumps
- Procedures for collecting documentation (SNA, TSO)
- VIT analysis tool
- Channel programs
- Flow diagrams
- Procedures for locating buffer pools
- CPCB operation codes
- Storage and control block ID codes
- Offset names and locations for SNA buffer pools.

Volume 2 includes information about the following:

- VIT entries
- SNA internal trace
- FFST dumps and probes

OS/390 SecureWay Communications Server: Data Areas Volume 1 (LY43-0111-03 [SC]). This book

describes SNA data areas and can be used to read a SNA dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

OS/390 SecureWay Communications Server: Data Areas Volume 2 (LY43-0112-03 [SC]). This book describes SNA data areas and can be used to read a SNA dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and Codes

OS/390 SecureWay Communications Server: SNA Messages (SC31-8569-03 [HC/SC]). This book describes the following types of messages and other associated information:

- Messages:
  - ELM messages for logon manager
  - IKT messages for TSO/SNA
  - IST messages for SNA network operators
  - ISU messages for sockets-over-SNA
  - IVT messages for the communications storage manager
  - IUT messages
  - USS messages
- Other information that displays in SNA messages:
  - Command and RU types in SNA messages
  - Node and ID types in SNA messages
- Supplemental message-related information

OS/390 SecureWay Communications Server: IP Messages Volume 1 (EZA) (SC31-8517-03 [HC/SC]). This volume contains TCP/IP messages beginning with EZA.

OS/390 SecureWay Communications Server: IP Messages Volume 2 (EZB) (SC31-8570-03 [HC/SC]). This volume contains TCP/IP messages beginning with EZB.

OS/390 SecureWay Communications Server: IP Messages Volume 3 (EZY-EZZ-SNM) (SC31-8674-03 [HC/SC]). This volume contains TCP/IP messages beginning with EZY, EZZ, and SNM.

OS/390 SecureWay Communications Server: IP and SNA Codes (SC31-8571-03 [HC/SC]). This book describes codes and other information that display in CS for OS/390 messages.

APPC Application Suite

OS/390 eNetwork Communications Server: APPC Application Suite User's Guide (GC31-8619-00 [SC]). This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC application suite. Although its primary audience is the end user, administrators and application programmers may also find it useful.

OS/390 eNetwork Communications Server: APPC Application Suite Administration (SC31-8620-00 [SC]). This book contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.

OS/390 eNetwork Communications Server: APPC Application Suite Programming (SC31-8621-00 [SC]). This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Multiprotocol Transport Networking (MPTN) Architecture Publications

Following are selected publications for MPTN:

Networking Blueprint Executive Overview (GC31-7057)

Multiprotocol Transport Networking: Technical Overview (GC31-7073)

Multiprotocol Transport Networking: Formats (GC31-7074)

Redbooks

The following Redbooks may help you as you implement CS for OS/390.

- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 1: Configuration and Routing* (SG24–5227–01).

  This book provides examples of how to configure the base TCP/IP stack, routing daemons and the TELNET server. This book

also provides information about national language support (NLS), routing, OSPF, network interfaces, diagnosis, multicasting, OS/390 UNIX System Services and security in an OS/390 UNIX System Services environment.

- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 2: UNIX Applications* (SG24–5228–01).

  This book provides information about implementing applications that run in the OS/390 UNIX environment, such as FTP, SNMP, BIND-based name server, DHCP, and SENDMAIL. This book also provides configuration samples and describes the implementation process.

- *OS/390 eNetwork Communication Server V2R7 TCP/IP Implementation Guide Volume 3: MVS Applications* (SG24–5229–01).

  This book provides information about TCP/IP applications that run in a legacy MVS environment, including CICS/IMS Sockets, and printing (NPF, LPR, and LPD.)

- *TCP/IP in a Sysplex* (SG24–5235–01).

  The main goals of a Parallel Sysplex are high availability and high performance. This book demonstrates how these goals can be achieved in the particular environment of SecureWay Communications Server for OS/390 and its TCP/IP applications. This book describes the WLM/DNS functions, the Network Dispatcher and the Dynamic VIPA.

- *SNA and TCP/IP Integration* (SG24–5291–00).

  This book provides information about integrating current SNA network with future TCP/IP and Web-based communication requirements. This book concentrates on routing techniques.

- *SNA in a Parallel Sysplex Environment* (SG24–2113–01).

  This book provides information about implementing a VTAM-based network on a Parallel Sysplex.

- *Subarea to APPN Migration : VTAM and APPN Implementation* (SG24–4656–01).

  This book is the first of two volumes. This book provides information about the migration of a subarea network to an APPN network. Some knowledge of SNA subarea networks and familiarity with the functions, terms and data flows of APPN networks is assumed.

- *Subarea to APPN Migration : HPR and DLUR Implementation* (SG24–5204–00).

  This book is the second of two volumes. This book provides information about the coverage of a network using HPR, DLUR and APPN/HPR routers. Some knowledge of SNA subarea networks and familiarity with the functions, terms and data flows of APPN networks is assumed.

Bibliography

Index

A

ACBNAME operand (APPL definition statement) 33
ACTIVATE operand
  dynamic reconfiguration 498
ADD statement (dynamic reconfiguration) 494
ADDR operand
  NCP major node 321
  PU definition statement
    dynamic reconfiguration 499
    model major node 261
    switched major node 411
address of channel-attached NCP 72
ADDRESS operand
  LINE definition statement
    channel-to-channel 72
    channel-to-NCP 72
adjacent cluster routing table
  defining 508
adjacent control point major node
  defining 19
  described 2
adjacent SSCP table 501
ADJCDRM definition statement
  adjacent SSCP table 503
ADJCP definition statement, adjacent control point major node 20
ADJNET operand
  GWPATH definition statement (CDRM major node) 118
ADJNETCS operand
  GWPATH definition statement (CDRM major node) 119
ADJNETEL operand
  GWPATH definition statement (CDRM major node) 119
ADJNETSA operand
  GWPATH definition statement (CDRM major node) 120
alias name translation
  coding APPL definition statement 29
ALL option (VTAM internal trace) 646
ALSLIST operand (CDRSC definition statement) 96
ALSREQ operand
  CDRSC definition statement 96
ALSREQ start option 560
ANS operand
  NCP definition statements
    takeover restriction 321
    VTAM restrictions on NCP definition statements 321
  PU (switched) definition statement
    description 411
    takeover restriction 262, 411
ANSWER operand 261
  LINE definition statement
    external communication adapter peripheral node 155
ANSWER operand 261 *(continued)*
  VTAM restrictions on NCP definition statements 322
API option (VTAM internal trace) 648
APPC operand
  APPL definition statement 34
APPC option (VTAM internal trace) 648
APPC trace 648
APPL definition statement
  alias translation facility 29
  coding 29
  logon manager 29
  TSO/VTAM 29
application program interface (API), internal trace of 648
application program major node
  defining 25
application programs
  LU 6.2 27
  non-LU 6.2 25
  registering 25
application-supplied dynamic parameter (ASDP) 412
APPLID operand
  LOGCHAR macroinstruction 695
  LOGOFF macroinstruction 763
  LOGON macroinstruction 759
APPN cross-domain resources 82
APPN-to-subarea class-of-service mapping table, defining 521
APPN transmission group profile (TGP)
  defining 513
  described 5
  IBM-supplied transmission group profiles 824
    IBMTGPS (member list containing the IBM-supplied TGP) 513
APPNCOS definition statement (APPN class of service) 668
APPNCOS operand
  MODEENT macroinstruction, logon mode table 702
APPNCOS start option 561
ASDP operand
  PU definition statement (switched major node) 412
ASIRFMSG start option 561
ASLENT macroinstruction 662
ASLENT operand
  APPL definition statement 34
  CDRSC definition statement 97
  LOCAL definition statement 193
  LU definition statement
    local SNA major node 210
    LU group major node 243
    model major node 262
    switched major node 412
  NCP definition statements 322
ASLPLU macroinstruction 662
ASLTAB macroinstruction 662
ASLTAB operand
  APPL definition statement 35
  CDRSC definition statement 97

© Copyright IBM Corp. 1984, 1999

ASLTAB operand *(continued)*
LOCAL definition statement 194
LU definition statement
local SNA major node 210
LU group major node 243
model major node 262
switched major node 412
VTAM restrictions on NCP definition statements 323
ASRCVLM operand 35
assembler features, restriction on use 15
associated LU table
description 661
installing 662
overview 6
ASYDE start option 562
ATNLOSS operand (APPL definition statement) 36
AUTH operand (APPL definition statement) 36
AUTHLEN operand 155
PU definition statement
local SNA major node 211
model major node 263
NCP major node 323
switched major node 413
AUTHLEN start option 562
AUTO operand
LINE definition statement (NCP definition statements) 323
AUTODL operand
NCP statements 324
AUTODMP operand
PCCU definition statement 324
AUTOIPL operand
PCCU definition statement 325
automatic disconnect (DISCNT), switched subarea 268, 422
automatic restart of NCP 325
AUTORTRY start option 562
AUTOSES operand
APPL definition statement 38
AUTOSYN operand (PCCU definition statement) 326
AUTOTI start option 563

B

BACKUP operand
PCCU definition statement 326
baseno (buffer pool start option) 567
BATCH operand
LU definition statement
model major node 263
switched major node 413
BCTC mode (basic channel-to-channel mode) 70
BHSET operand (VTAM restrictions on NCP definition statements 327
BN start option 564
BNDYN operand
NETWORK definition statement 511
BNDYN start option 564
BNORD start option 565
border node COS mapping table 526
BSCMDRS start option 565
BSCTMOUT start option 566
BUFCAP operand
LINE definition statement
NCP major node 327
BUFFER operand (USSMSG macroinstruction) 739
buffer pool start option 566
buffer pool start option format 566
BUFNUM operand 194
bufsize (buffer pool start option) 567
BUILD definition statement 297

C

CACHETI start option 572
CALL operand
LINE definition statement
external communication adapter peripheral node 158
PATH definition statement (switched major node) 414
security device 328
VTAM restrictions on NCP definition statements 328
CAPACITY operand
CDRM definition statement 120
LINEROW definition statement
APPN class of service 670
PORT definition statement
XCA major node 158
PU definition statement
local SNA major node 211
model major node 264
NCP major node 328
switched major node 414
TGP definition statement
transmission group profile 515
CDRDYN operand
CDRM definition statement (CDRM major node) 120
CDRDYN start option 572
CDRM (cross-domain resource manager) major node
defining in interconnected SNA network 113
defining in multiple-domain network 113
CDRM definition statement
adjacent SSCP table 502
CDRM major node 116
CDRM operand
CDRSC statement (CDRSC major node) 98
CDRSC (cross-domain resource), defining 81
CDRSC definition statement (CDRSC major node) 93
CDRSC major node, defining 81
CDRSC operand
CDRM definition statement (CDRM major node) 121
CDRSCTI start option 573
CDSERVR start option 574
CDSREFER start option 574
CDUMPDS operand (PCCU definition statement) 329
central directory server 574
CERTIFY operand
APPL definition statement 39
LU definitiomn statement
local SNA major node 212
LU group major node 244

CERTIFY operand *(continued)*
model major node 264
NCP major node 330
switched major node 415
CHANCON operand
PCCU definition statement 329
PU definition statement (channel-attached NCP) 72
change number of sessions (CNOS) request
DDRAINL operand 40
DMINWNL operand 41
DMINWNR operand 41
DRESPL operand 42
DSESLIM operand 42
channel-attached NCP support, defining 67
channel-attachment major node, defining 65
channel contact request
conditional 329
defining type of 72, 329
unconditional 72, 329
channel I/O, VTAM internal trace of 648
channel link name 72
channel link station name (PCCU definition statement)
DUMPSTA operand 339
LOADSTA operand 355
character-coded commands
conversion examples 833
logon 733
order of USS table use 733
syntax 757
CINDXSIZ start option 574
CKEYNAME operand 212, 244, 265, 415, 702
LU definition statement 330
class of service (COS), APPN 672
APPNCOS definition statement
PRIORITY operand 674
default values for APPN classes of service 679
definitions 668
IBM-supplied APPN classes of service
COSAPPN (member list containing the IBM-supplied COS) 680
definition list 798
described 680
implementing APPN classes of service 669
LINEROW definition statement 672
CAPACITY operand 670
COSTBYTE operand 671
COSTTIME operand 671
NUMBER operand 673
PDELAY operand 674
SECURITY operand 676
UPARM1 operand 676
UPARM2 operand 676
UPARM3 operand 677
WEIGHT operand 677
NODEROW definition statement
CONGEST operand 670
NUMBER operand 673
ROUTERES operand 675
WEIGHT operand 678
class of service (COS), subarea
SSCP (ISTVTCOS) 665
class of service (COS), subarea *(continued)*
table
described 664
unnamed entry 665
CLRSESSQ operand
local SNA major node 212
Model major node 265
Switched major node 416
VTAM restrictions on NCP definition statements 331
CLUSTER definition statement
NCP operands used by VTAM 301
CMD operand (USSCMD macroinstruction) 740
CMIP, VTAM internal trace of 648
CMPAPPLI operand
APPL definition statement 39
CMPAPPLO operand
APPL definition statement 39
CMPMIPS start option 575
CMPVTAM start option 575
CNMTAB start option 576
CNOS (change number of sessions) request (APPL definition statement)
DDRAINL operand 40
DMINWNL operand 41
DMINWNR operand 41
DRESPL operand 42
DSESLIM operand 42
coding conventions for definition statements 9
coding formats for macroinstructions 10
COLD or WARM start option 576
comments in VTAM macroinstructions 12
communication network management (CNM) application 725
COMPRES operand (MODEENT macroinstruction, logon mode table) 703
COMPROT operand (MODEENT macroinstruction, logon mode table) 703
COMRATE operand
NCP major node 307
switched major node 398, 400
CONFGDS operand
LBUILD definition statement 195
PCCU definition statement 331
VBUILD definition statement
CDRM major node 122
CDRSC major node 99
channel-attachment major node 73
local SNA major node 213
switched major node 416
CONFGPW operand
LBUILD definition statement 195
PCCU definition statement 331
VBUILD definition statement
CDRM major node 122
CDRSC major node 100
channel-attachment major node 73
local SNA major node 213
switched major node 417
CONFIG start option 576
CONGEST operand (APPN class of service) 670

connection network
VNGROUP operand
NCP major node 387
XCA major node 187
VNNAME operand
NCP major node 387
XCA major node 187
CONNTYPE operand 159
PU definition statement
local SNA major node 213
model major node 266
NCP major node 331
switched major node 417
CONNTYPE start option 577
continuation character 12
COPIES operand (NETWORK definition statement) 332
COS (class of service), APPN
APPNCOS definition statement
PRIORITY operand 674
default values for APPN classes of service 679
definitions 668
IBM-supplied APPN classes of service
COSAPPN (member list containing the IBM-supplied COS) 680
definition list 798
described 680
implementing APPN classes of service 669
LINEROW definition statement 672
CAPACITY operand 670
COSTBYTE operand 671
COSTTIME operand 671
NUMBER operand 673
PDELAY operand 674
SECURITY operand 676
UPARM1 operand 676
UPARM2 operand 676
UPARM3 operand 677
WEIGHT operand 677
NODEROW definition statement
CONGEST operand 670
NUMBER operand 673
ROUTERES operand 675
WEIGHT operand 678
COS (class of service), subarea
SSCP (ISTVTCOS) 665
table
described 664
unnamed entry 665
COS macroinstruction 665
COS operand
MAPSTO definition statement
APPN-to-subarea COS mapping definitions 523, 525
border node COS mapping definitions 528
MODEENT macroinstruction 704
COSEND macroinstruction 665
COSTAB macroinstruction 665
COSTAB operand
BUILD definition statement (NCP major node) 333
COSTAB operand *(continued)*
NETWORK definition statement (NCP major node) 333
COSTBYTE operand
CDRM definition statement 122
LINEROW definition statement
APPN class of service 671
PU definition statement
local SNA major node 214
model major node 267
NCP major node 333
switched major node 418
TGP definition statement
transmission group profile 518
COSTTIME operand
CDRM definition statement 123
LINEROW definition statement
APPN class of service 671
PU definition statement
local SNA major node 215
model major node 267
NCP major node 333
switched major node 418
TGP definition statement
transmission group profile 519
CPCDRSC start option 578
CPCP operand 160
PU definition statement
local SNA major node 215
model major node 267
NCP major node 334
switched major node 418
CPCP start option 578
CPNAME operand 161
CDRSC definition statement (cross-domain resource major node) 100
NEXTCP definition statement 511
PU definition statement
local SNA major node 215
NCP major node 334
switched major node 419
cross-domain resource manager (CDRM) major node
defining in interconnected SNA network 113
defining in multiple-domain network 113
CSA24 start option 580
CSALIMIT start option 579
CTCA, defining 65
CUADDR operand
LOCAL definition statement (local non-SNA major node) 195
PCCU definition statement 334
PU definition statement (local SNA major node) 216

D

DATA operand (LOGON command) 759
DATEDLM operand (USSTAB macroinstruction, USS table) 741
DATEFORM start option 580
DATEFRM operand (USSTAB macroinstruction, USS table) 741

DATMODE operand
PU definition statement
model major node 268
switched major node 420
VTAM restrictions on NCP definition statements 335
DBCS (double-byte character set) support (MODEENT macroinstruction) 707
DCODE operand (MODEENT macroinstruction) 704
DDRAINL operand (APPL definition statement) 40
DEFAULT operand
MAPSTO definition statement 523, 525
USSPARM macroinstruction 742
default SSCP lists, defining 502
default USS table
operation-level 785
session-level 784
defaults, using 13
defining major nodes
adjacent control point 19
application program 25
CDRM 113
CDRSC 81
channel-attachment 65
external communication adapter 133
local non-SNA 191
local SNA 202
LU group 240
model 253
NCP 295
switched 392
DELAY operand
PCCU definition statement (NCP major node) 335
PORT definition statement (XCA major node) 162
PU definition statement
channel-to-channel 73
channel-to-NCP 73
local SNA major node 216
DELETE statement (dynamic reconfiguration) 495
DELETER operand
NCP dynamic path update 486
VTAM dynamic path update 491
VTAM PATH definition statement 482
DESC operand (USSMSG macroinstruction) 742
description of VTAM major nodes 2
description of VTAM tables and filter 6
DESTSA operand
dynamic path update
NCP 487
VTAM 491
PATH definition statement 482
device number of channel-attached NCP 72
DIAL operand
GROUP definition statement
external communication adapter major node 162
VTAM restrictions on NCP definition statements 336
DIALNO operand
PATH definition statement (switched major node) 420
DIALRTRY start option 581
directory definition file for CMIP services, coding 681
DIRSIZE start option 581
DIRTIME start option 582
DISCNT operand
PU definition statement
local SNA major node 216
model major node 268
switched major node 422
VTAM restrictions on NCP definition statements 336
DISCNTIM start option 582
DISJOINT operand
CDRM definition statement (cross-domain resource manager major node) 123
DLCADDR operand 162
PATH definition statement 423
DLCI operand 337
NCP major node 307
switched major node 398, 400
DLOGMOD operand
APPL definition statement 40
CDRSC definition statement 100
LOCAL definition statement 195
LU definition statement
local SNA major node 217
LU group major node 245
model major node 269
switched major node 429
VTAM restrictions on NCP definition statements 337
DLRORDER start option 583
DLRTCB start option 583
DLURNAME operand
PU definition statement 429
DMINWNL operand (APPL definition statement) 41
DMINWNR operand (APPL definition statement) 41
domain
defining multiple to VTAM 1
defining single to VTAM 1
DRESPL operand (APPL definition statement) 42
DRTYPE operand (DR ADD definition statement) 499
DSESLIM operand (APPL definition statement) 42
DSPLYDEF start option 584
DSPLYMAX start option 584
DSPLYWLD operand (APPL definition statement) 43
DSPLYWLD start option 585
DSPSIZE start option 645
dump communication controller
after failure 324
automatic 324
communication controller dump file 338
CSP dump file 329
MOSS dump file 362
dump station name (DUMPSTA) 339
DUMPDS operand (PCCU definition statement) 338
DUMPLD operand (PCCU definition statement) 339
DUMPSTA operand (PCCU definition statement) 339
DUPDEFS start option 585
DWACT operand (PU definition statement, switched major node) 430
DYNADJCP operand
PU definition statement
local SNA major node 217
model major node 269
NCP major node 340

DYNADJCP operand *(continued)*
switched major node 430
DYNADJCP start option 586
dynamic path update
description 485
NCP 485
VTAM 490
dynamic reconfiguration
active and inactive requirements 494
ADD definition statement 494
adding a PU or LU to a line 494
adding an LU to a PU 494
DELETE definition statement 495
deleting a PU and its LUs 495
deleting an LU 495
MOVE definition statement 496
moving a PU 496
dynamic reconfiguration and dynamic change 494
DYNASSCP start option 587
DYNDLGMD start option 587
DYNHPPFX start option 588
DYNLU operand 167
ADJCP definition statement, ADJCP major node 21
PU definition statement
local SNA major node 218
model major node 271
switched major node 431
VTAM restrictions on NCP definition statements 340
DYNLU start option 588
DYNMODTB start option 589
DYNPU operand
GROUP definition statement
NCP major node 341
DYNPUPFX operand
GROUP definition statement
NCP major node 342
DYNPUPFX start option 589
DYNVNPFX start option 589
DYNWIND operand
NCP major node 307
switched major node 398, 400

E

EAS operand
APPL definition statement 43
CDRSC definition statement 101
LU definition statement
local SNA major node 219
ECLTYPE operand
VTAM restrictions on NCP definition statements 342
ELEMENT operand (CDRM major node)
CDRM definition statement 124
GWPATH definition statement 124
EN (end node) 596
ENCR operand
APPL definition statement 44, 101
LU definition statement
local SNA major node 219
LU group major node 245
model major node 271
ENCR operand *(continued)*
LU definition statement *(continued)*
NCP major node 343
switched major node 432
MODEENT macroinstruction 704
ENCRPREF start option 590
ENCRTYPE operand
application program major node 45
cross domain resource major node 102
local sna major node 220
lu group major node 246
model major node 271
ncp major node 343
switched major node 432
ENCRYPTN start option 590
END macroinstruction 727
ENDINTAB macroinstruction 691
ENHADDR start option 591
Enterprise Extender
external communication adapter major node 136, 151, 171, 172, 179, 181
switched major node 396, 408, 422, 436, 455
ERn operand
NCP dynamic path update 487
PATH definition statement 483
VTAM dynamic path update 492
ESC option (VTAM internal trace) 648
ESIRFMSG start option 592
extended recovery facility (XRF)
ANS operand
model major node 262
NCP major node 321
switched major node 411
HAVAIL operand (application program major node) 47
EXTRN definition statement 751

F

F parameter, buffer pool start option 568
FASTPASS operand
APPL definition statement 46
LU definition statement
local SNA major node 221
LU group major node 246
model major node 272
switched major node 433
VTAM restrictions on NCP definition statements 344
FEATUR2 operand
LOCAL definition statement 196
LU definition statement
model major node 273
switched major node 434
VTAM restrictions on NCP definition statements 345
filing VTAM network definition statements 18
FLDEND macroinstruction 714
FLDENT macroinstruction 714
FLDTAB macroinstruction 714
FLDTAB start option 593
FMPROF operand (MODEENT macroinstruction) 706
FORMAT operand
USSCMD macroinstruction 743

FORMAT operand *(continued)*
USSTAB macroinstruction 743
frame relay 346
FRELAY operand (NCP major node) 346
FROM operand
DELETE (DR) definition statement 499
MOVE (DR) definition statement 499
FRSESET definition statement (NCP major node) 299
FSIRFMSG start option 593

G

gateway paths 116
general trace facility (GTF) 645
GID operand (switched PATH definition statement) 435
GP3174 operand
description 347
VTAM restrictions on NCP definition statements 347
GPOLL operand
VTAM restrictions on NCP definition statements 347
GROUP definition statement
channel-attachment major node
channel-to-channel adapter 65
channel-to-NCP link 67
multipath channel (MPC) support 66
cross-domain resource major node 93
external communication adapter major node
external communication adapter peripheral node 136
external communication adapter subarea node 134
NCP major node 300
GRPNM operand
PATH definition statement
switched major node 435
GWCTL operand (PCCU definition statement) 348
GWN operand (CDRM major node)
GWPATH definition statement 124
GWNAU definition statement (NCP) restrictions 299
GWPATH definition statement (CDRM major node) 116
GWSELECT operand (CDRM definition statement) 125
GWSSCP start option 594

H

HARDCOPY operand (FLDENT macroinstruction) 717
HAVAIL operand (APPL definition statement) 47
HNTSIZE start option 594
HOLD operand (LOGOFF command) 763
HOST definition statement
considerations when defining channel-attached major node 69
VTAM restrictions in NCP 297
HOSTNAME 436
HOSTPU start option 595
HOSTSA start option 595
HOTIOTRM start option 596
HPR operand
PU definition statement 169
local SNA major node 125, 221
NCP major node 348
switched major node 436
HPR option (VTAM internal trace) 648
HPR start option 597
HPRPST start option 598
HPRQLIM operand
NCP major node 307
switched major node 398, 400
HSRTSIZE start option 599

I

IBMTEST command 765
ICN (interchange network node) 596
IDBLK operand (PU definition statement)
switched major node 437
IDNUM operand (PU definition statement)
switched major node 437
INBFRS operand (HOST definition statement) 349
INCLUD0E 221, 274, 349, 438
independent LU cross-domain resource, defining 81
INITDB start option 600
INITEST operand (PCCU definition statement) 349
INNPORT operand
VTAM restrictions on NCP definition statements 350
INOPDUMP start option 600
INTAB macroinstruction 691
interpret table
described 690
installing and changing 693
logon-interpret routines, coding 692
logon message, defining 691
INTFACE operand
NCP major node 307
switched major node 398, 400
IOINT start option 600
IOMSGLIM start option 601
IOPURGE start option 601
IPADDR start option
start option definition 602
switched major node 439
syntax diagram 534
IPPORT 169
IPQLIM operand
NCP major node 307
switched major node 398, 400
IPRESOLV operand 439
IPTOS 169
IRETRY operand
PU definition statement
model major node 274
switched major node 439
IRNSTRGE start option 602
ISTADJCP
dynamic adjacent control point major node 19
ISTAPNCP
generic representation for an APPN CDRM 505
ISTAPNPU
generic representation for an APPN adjacent link station 96
ISTATUS operand
CDRM definition statement (CDRM major node) 126

ISTATUS operand *(continued)*
CDRSC definition statement (CDRSC major node) 103
GROUP definition statement
CDRSC major node 103
channel-to-channel 74
channel-to-NCP 74
external communication adapter major node 170
multipath channel (MPC) 74
LINE definition statement
channel-to-channel 74
channel-to-NCP 67
external communication adapter major node 170
multipath channel (MPC) 74
LOCAL definition statement 197
LU definition statement
local SNA major node 222
switched major node 440
PU definition statement
channel-attached NCP 74
channel-to-channel 74
external communication adapter major node 170
local SNA major node 222
multipath channel (MPC) 74
switched major node 440
VTAM restrictions on NCP definition statements 351
ISTCOSDF, default COS entry in default logon mode table 665
ISTCOSDF start option 603
ISTGENERIC, IBM-supplied generic resources structure name 634
ISTINCDT, session-level USS table
discussed 732
listed 784
ISTINCLM, default logon mode table
described 698
listed 769
ISTINCNO, operation-level USS table
contents 732
listed 785
ISTMGC10, default session awareness data filter 725
ISTPDILU, file for independent LUs 85
ISTSDCOS, assembled COS table 667
ISTVTCOS, the SSCP COS table entry 665

K

kanji character set support 707
katakana character set support 707
KCLASS macroinstruction 726
KCLASS operand (MAPSESS macroinstruction) 727
KEEPMEM macroinstruction 726

L

LADDR operand
NCP major node 307
switched major node 398, 400
LANG operand
IBMTEST command 766
LOGOFF command 763
LOGON command 759
LANG operand *(continued)*
MODEENT macroinstruction 706
LANGTAB operand
IBMTEST command 766
LOGOFF command 764
LOGON command 760
LBUILD definition statement 192
LCS option (VTAM internal trace) 649
LIMINTCP start option 603
LIMQSINT operand (APPL definition statement) 47
LIMRES operand
LINE definition statement
external communication adapter peripheral node 170
PU definition statement
local SNA major node 222
model major node 275
switched major node 440
VTAM restrictions on NCP definition statements 352
LINE definition statement
channel-attachment major node
channel-to-channel adapter 67
channel-to-NCP 67
external communication adapter major node
external communication adapter peripheral node 136
external communication adapter subarea node 134
NCP major node 300
LINEAUT operand (NCP) definition statement, VTAM restrictions on 352
LINENM operand (PATH definition statement)
switched major node 441
LINEROW definition statement (APPN class of service) 668
link station for dumps 339
LIST operand (message-flooding prevention table) 718
LIST start option 604
LISTBKUP start option 604
LIVTIME 171
LLERP operand
PU definition statement
NCP major node 352
switched major node 441
LMDENT operand (APPL definition statement) 47
LMODADR operand
VTAM restrictions on NCP definition statements 353
LNCTL operand
GROUP definition statement
channel-attached NCP 74
channel-to-channel attachment 74
multipath channel (MPC) 74
TRLE definition statement 472
VTAM restrictions on NCP definition statements 354
LOADFROM operand (PCCU definition statement) 354
LOADLIB operand (BUILD definition statement) 355
LOADSTA operand (PCCU definition statement) 355
LOCADDR operand
LU definition statement
local SNA major node 223
model major node 276

LOCADDR operand *(continued)*
switched major node 442
VTAM restrictions on NCP definition statements 356
LOCAL definition statement 192
local non-SNA major node, defining 191
local SNA major node, defining 202
LOGAPPL operand
CDRSC definition statement 103
LOCAL definition statement 197
LU definition statement
local SNA major node 224
LU group major node 247
model major node 276
switched major node 442
VTAM restrictions on NCP definition statements 357
LOGCHAR macroinstruction 691
LOGMODE operand
LOGON command 761
MODEENT macroinstruction 707
LOGOFF command 762
LOGON command 758
logon manager, defining to VTAM 29
logon mode table
defined 698
installing 700
MVS defaults listed 769
session parameters 699
LOGTAB operand
LOCAL definition statement 198
LU definition statement
local SNA major node 224
LU group major node 248
model major node 277
switched major node 443
VTAM restrictions on NCP definition statements 357
LOSTERM exit routine 764
LOSTERM operand (APPL definition statement) 48
LPDA operand
VTAM restrictions on NCP definition statements 358
LU definition statement
local SNA major node 204
LU group major node 241
model major node 255
switched major node 397, 399
LU definition statement (NCP) operands used by VTAM 301
LU group major node, defining 240
LUALIAS operand
CDRSC definition statement (cross-domain resource major node) 104
LUALIAS values after a merge 85
LUAPFX operand 48
LUDR operand
local SNA major node 224
LU group major node 248
NCP major node 358
LUDRPOOL (NCP) definition statement 298
LUGROUP definition statement 241
LUGROUP operand
PU definition statement
local SNA major node 225
LUGROUP operand *(continued)*
PU definition statement *(continued)*
model major node 277
switched major node 443
VTAM restrictions on NCP definition statements 358
LUNAME argument of TEXT operand 752
LUSEED operand
PU definition statement
local SNA major node 225
model major node 277
switched major node 443
VTAM restrictions on NCP definition statements 359

M

MAC operand 49, 708
MACLNTH operand 49, 709
MACTYPE operand 50, 709
MAINTLVL start option 605
major nodes, defining
adjacent control point 19
application program 25
CDRM 113
CDRSC 81
channel-attachment 65
external communication adapter 133
local non-SNA 191
local SNA 202
LU group 240
model 253
NCP 295
switched 392
MAPSESS macroinstruction 726
MAPSTO definition statement 528
MAXBFRU operand
HOST definition statement 359
LINE definition statement
channel-to-channel 75
channel-to-NCP 75
multipath channel (MPC) 75
PU definition statement
local SNA major node 226
TRLE definition statement 472
MAXDATA operand
LOCAL SNA definition statement 226
PCCU definition statement 360
PU definition statement
channel-attached NCP 76
model major node 278
switched major node 444
MAXDLUR operand (VBUILD definition statement) 445
MAXGRP operand (VBUILD definition statement)
switched major node 445
MAXLOCAT start option 605
MAXLURU start option 606
MAXNO operand (VBUILD definition statement)
switched major node 445
MAXOUT operand
PU definition statement
model major node 279
switched major node 446

MAXOUT operand *(continued)*
VTAM restrictions on NCP definition statements 360
MAXPVT operand (APPL definition statement) 50
MAXREADS operand 473
MAXSESS operand
CDRSC definition statement 105
LU definition statement
NCP major node 360
switched major node 446
PU definition statement
model major node 279
MAXSSCPS start option 606
MAXSUBA operand
BUILD definition statement 361
NETWORK definition statement 361
MAXSUBA start option 606
MCSFLAG operand (USSMSG macroinstruction) 745
MDH (migration data host) 596
MDLENT macroinstruction 721
MDLENT operand
APPL definition statement 51
CDRSC definition statement 106
LOCAL definition statement 198
LU definition statement
local SNA major node 227
LU group major node 248
model major node 280
switched major node 447
VTAM restrictions on NCP definition statements 361
MDLPLU macroinstruction 721
MDLTAB macroinstruction 721
MDLTAB operand
APPL definition statement 51
CDRSC definition statement 106
LOCAL definition statement 199
LU definition statement
local SNA major node 228
LU group major node 249
model major node 280
switched major node 447
VTAM restrictions on NCP definition statements 361
MDUMPDS operand (PCCU definition statement) 362
merged resource definitions
LUALIAS values after a merge 85
NQNMODE values after a merge 87
rules for 83
message-flooding prevention table
description 714
installing 714
overview 7
MESSAGE operand (FLDENT macroinstruction) 719
messages, VTAM internal trace of 649
MIH operand
LINE definition statement (CTCA) 77
MIHTMOUT start option 607
MLTGPRI operand 447
MODEEND macroinstruction (logon mode table) 699
MODEENT macroinstruction (logon mode table)
description 699
model LU 240
model LU group 240
model major node, defining 253
model name table
description 720
installing 721
overview 8
MODEL operand
LOGON command 761
MDLENT macroinstruction 722
MDLPLU macroinstruction 722
MODETAB macroinstruction (logon mode table) 699
MODETAB operand
APPL definition statement 52
CDRSC definition statement 106
LOCAL definition statement 199
LU definition statement
local SNA major node 228
LU group major node 249
model major node 280
switched major node 448
VTAM restrictions on NCP definition statements 362
MOVE definition statement (dynamic reconfiguration) 496
MPC support, channel-attachment major node 66
MPCLEVEL operand 474
MPCUSAGE operand 474
MSG operand (USSMSG macroinstruction)
description 745
terminal operator messages 745
VTAM operator messages 745
MSGLEVEL start option 607
MSGLVL operand (USSMSG macroinstruction, USS table)
description 745
messages supported by MSGLVL 746
MSGMOD start option 608
MULTGSUP operand 174
multiple definition of resources 83
MXSAWBUF start option 609
MXSSCPRU start option 609
MXSUBNUM start option 609

N

name field
VTAM definition statements 9
VTAM macroinstructions 10
names, VTAM restriction on 14
names to avoid for nodes 14
national characters 9
native ATM connections 138
NATIVE operand 174
ADJCP definition statement 23
PU definition statement 229
model major node 281
switched major node 448
NCP (Network Control Program)
generation
definition statements used by VTAM 296
operands used by VTAM 301
major node, defining 295
PCCU definition statement 296
resources controlled by 300

NCP (Network Control Program) *(continued)*
VTAM restrictions on
BUILD definition statement 297
GWNAU definition statement 299
HOST definition statement 297
LUDRPOOL definition statement 298
NETWORK definition statement 298
PUDRPOOL definition statement 298
SYSCNTRL definition statement 298
NCP definition statements
BUILD definition statement 297
coding VTAM-only operands 321
GWNAU definition statement 299
HOST definition statement 297
LUDRPOOL definition statement 298
NETWORK definition statement 298
PCCU definition statement 296
PUDRPOOL definition statement 298
SYSCNTRL definition statement 298
NCPBUFSZ start option 610
NCPPATH definition statement (dynamic path update) 485
NETID 175
NETID operand
ADJCP definition statement, adjacent control point major node 23
BUILD definition statement 364
HOST definition statement 364
NCPPATH definition statement 488
NETSRVR definition statement
network node server list 531
NETWORK definition statement
adjacent cluster routing table 512
adjacent SSCP table 507
CDRM major node 127
CDRSC major node 107
COS mapping table 529
NCP major node 364
PCCU definition statement 364
PU definition statement
local SNA major node 229
model major node 282
switched major node 449
VPATH definition statement 492
VTAM restrictions on NCP definition statements 364
NETID start option 610
NETSRVR definition statement, network node server list 530
NETSRVR operand
CDRSC definition statement (cross-domain resource major node) 107
Network Control Program (NCP)
generation
definition statements used by VTAM 296
operands used by VTAM 301
major node, defining 295
PCCU definition statement 296
resources controlled by 300
VTAM restrictions on
BUILD definition statement 297
GWNAU definition statement 299
Network Control Program (NCP) *(continued)*
VTAM restrictions on *(continued)*
HOST definition statement 297
LUDRPOOL definition statement 298
NETWORK definition statement 298
PUDRPOOL definition statement 298
SYSCNTRL definition statement 298
NETWORK definition statement
adjacent cluster routing table 509
adjacent SSCP table 502
border node COS mapping table 527
CDRM major node 115
CDRSC major node 93
NCP major node 298
network node server list (NNSL)
defining 5
described 529
NEWNAME operand (BUILD definition statement) 365
NEWPATH operand (PCCU definition statement) 365
NEXTCP definition statement
adjacent cluster routing table 509
NMVTLOG start option 610
NN (network node) 596
NN operand 176
ADJCP definition statement (adjacent control point major node) 24
CDRM definition statement 127
PU definition statement
local SNA major node 230
model major node 282
NCP major node 366
switched major node 450
NNSL (network node server list)
defining 5
described 529
NNSPREF start option
start option definition 611
syntax diagram 534
node types and their functions 596
NODELST start option 612
NODEROW definition statement (APPN class of service) 669
NODETYPE start option 612
NONE option, VTAM internal trace 649
nonnative network attachment (XNETALS start option) 657
NOPROMPT or PROMPT start option 618
NOTNSTAT or TNSTAT start option 637
NOTRACE operand, VTAM internal trace 646
NQNMODE operand
CDRSC definition statement (cross-domain resource major node) 108
NQNMODE values after a merge 87
NQNMODE start option 613
NRM option, VTAM internal trace 649
NSEXIT exit routine 765
NSRTSIZE start option 613
NUMBER operand (APPN class of service)
LINEROW definition statement 673
NODEROW definition statement 673
NUMTREES start option 614

O

operands field
  VTAM definition statements 9
  VTAM macroinstructions 11
operation field of VTAM macroinstructions 11
operation-level USS table 732
operator commands
  terminal 757
  VTAM 756
OPERCNOS operand (APPL definition statement) 54
OPT=BLKSUP operand (USSMSG macroinstruction) 748
OPTION operand, VTAM internal trace
  ALL 646
  API (application program interface) 648
  APPC (LU 6.2 communication) 648
  CIO (channel I/O) 648
  CMIP (Common Management Information Protocol) 648
  default 648
  ESC (execution sequence control) 648
  HPR (high performance routing) 648
  LCS (LAN Channel Station) 649
  LOCK (locking) 649
  MSG (messages) 649
  NONE (no traces) 649
  NRM (network resource management) 649
  PIU (path information unit) 649
  PSS (process scheduling services) 649
  SMS (storage management services) 649
  SSCP (system services control point) 649
  VCNS (VTAM Common Network Services) 649
order of USS table use 733
ORDER operand
  VBUILD definition statement (network node server list) 531
OSA-Express
  transport resource list major node 469, 470, 472, 474, 475, 476, 478
OSIEVENT start option 614
OSIMGMT start option 615
OSITOPO start option 616
OSRTSIZE start option 616
outbound calls, allocating 414
overriding sift-down value 13
OWNER operand
  NCP definition statements, VTAM restrictions on 366
  PCCU definition statement 367

P

PACING operand
  CDRSC definition statement 109
  LU definition statement
    local SNA major node 230
    LU group major node 250
    model major node 283
    switched major node 450
  VTAM restrictions on NCP definition statements 367
PARM operand (USSPARM macroinstruction) 748
PARSESS operand (APPL definition statement) 55
PASSLIM operand
  PU definition statement
    model major node 283
    switched major node 451
PATH definition statement
  NCP routes, dynamic update of 486
  switched major node 399
  VTAM routes
    defining 481
    dynamic update of 490
path information unit (PIU), VTAM internal trace of 649
PCCU statement
  coding 296
  gateway control functions 348
  purpose of 296
PDELAY operand
  CDRM definition statement 128
  LINEROW definition statement
    APPN class of service 674
  PORT definition statement
    XCA major node 176
  PU definition statement
    local SNA major node 231
    model major node 284
    NCP major node 368
    switched major node 451
  TGP definition statement
    transmission group profile 519
PDTRCBUF start option 617
PERSIST operand
  APPL definition statement (application program major node 55
PHYSRSC operand
  VTAM restrictions on NCP definition statements 369
PID operand (switched PATH definition statement) 452
PIUMAXDS start option 617
PL/I programming language syntax 743
PLU operand
  ASLUPLU macroinstruction 663
  MDLPLU macroinstruction 723
PLUALMSG start option 617
POAQLIM operand
  APPL definition statement 56
PORTNAME operand 177, 475
positional parameters 744
PPOLOG start option 618
PRI operand (MAPSESS macroinstruction) 728
PRINET operand (MAPSESS macroinstruction) 728
PRINTER1 operand
  ASLENT macroinstruction 663
  ASLPLU macroinstruction 663
  LOGON command 761
PRINTER2 operand
  ASLENT macroinstruction 664
  ASLPLU macroinstruction 664
  LOGON command 761
PRIORITY operand (APPN class of service) 674
PRIPROT operand (MODEENT macroinstruction) 710
PROMPT or NOPROMPT start option 618

PRTCT operand
APPL definition statement 57
PU definition statement (switched major node) 452
PSERVIC operand
MODEENT macroinstruction 710
PSNDPAC operand (MODEENT macroinstruction) 710
PSS option, VTAM internal trace 649
PSSTRACE start option 619
PU definition statement
channel-attachment major node
channel-attached NCP 68
channel-to-channel adapter 65
external communication adapter major node
peripheral 136
subarea 134
local SNA major node 204
model major node 255
NCP major node 301
switched major node 396, 399
PU operand
VTAM restrictions on NCP definition statements 370
PUDR operand
NCP major node 371
PUDRPOOL (NCP) definition statement 298
PUNAME operand (BUILD definition statement) 371
PUTYPE operand
PU definition statement
channel-attached NCP 78
channel-to-channel 78
external communication adapter peripheral node 178
local SNA major node 231
model major node 285
multipath channel (MPC) 78
switched major node 453
VTAM restrictions on NCP definition statements 372
PVCNAME operand 178

R

railroad track syntax diagrams, how to read 15
RAR (route addition resistance) 675
READ operand 78
LINE definition statement (channel-attachment major node) 78
TRLE definition statement 476
reconfiguration, dynamic
active and inactive requirements 494
ADD definition statement 494
adding a PU or LU to a line 494
adding an LU to a PU 494
DELETE definition statement 495
deleting a PU and its LUs 495
deleting an LU 495
MOVE definition statement 496
moving a PU 496
RECOVERY operand
CDRM definition statement (CDRM major node) 128
REDIAL operand (PATH definition statement)
switched major node 453
REGISTER operand
APPL definition statement (application program major node) 57
CDRSC definition statement (cross-domain resource major node) 110
LU definition statement
local SNA major node 232
LU group major node 251
model major node 285
switched major node 454
PU definition statement
model major node 285
NCP major node 372
registering application programs 25
registering independent LUs 82
registering resources 25
REMOVE operand 696
REP operand
USSCMD macroinstruction 749
USSPARM macroinstruction 749
REPLYTO operand
GROUP definition statement
channel-to-channel 79
multipath channel (MPC) 79
TRLE definition statement 477
RESETLU operand 232, 251, 286, 373, 454
RESSCB operand
CDRSC definition statement 110
LU definition statement
switched major node 455
NCP definition statements, VTAM restrictions on 373
RESUSAGE start option 620
RETRIES operand
PU definition statement
NCP major node 373
RNAME operand
PCCU definition statement 374
ROUTCDE operand (USSMSG macroinstruction) 750
ROUTERES operand (APPN class of service) 675
ROUTERES start option 621
routes, defining 481
RUSIZES operand (MODEENT macroinstruction) 711

S

SACONNS start option 621
SALIMIT operand
BUILD definition statement 375
NETWORK definition statement 375
SAMAP 723
SAPADDR 455
SAPADDR operand
PU definition statement
external communication adapter major node 179
SAVEMOD operand (PCCU definition statement) 375
SAVERSCV
start option definition 621
syntax diagram 534
SAW operand (KCLASS macroinstruction) 728
SAWMAXDS start option 623
SAWMXQPK start option 623

SBCS (single-byte character set) support 707
SDLCMDRS start option 623
SEC operand (MAPSESS macroinstruction) 729
SECACPT operand
  APPL definition statement 58
SECLVL operand
  APPL definition statement 58
SECLVLCP start option 624
SECNET operand
  MAPSESS macroinstruction 729
  NCP definition statements, VTAM restrictions on 376
  PU definition statement
    local SNA major node 233
    model major node 286
    switched major node 456
SECPROT operand (MODEENT macroinstruction) 712
security, session
  operands affecting
    ENCR 44, 101, 219, 245, 271, 343, 432, 704
    SECLVL 58
    VERIFY 63, 465
  start options affecting 553
SECURITY operand
  CDRM definition statement 129
  LINEROW definition statement
    APPN class of service 676
  PORT definition statement
    XCA major node 180
  PU definition statement
    local SNA major node 233
    model major node 287
    NCP major node 376
    switched major node 456
  TGP definition statement
    transmission group profile 520
SEGMENT operand
  VTAM restrictions on NCP definition statements 377
SEQNCE operand (LOGCHAR macroinstruction) 697
session awareness (SAW) data filter 725
session-level USS table 731
session parameters 698
session security
  operands affecting
    ENCR 44, 101, 219, 245, 271, 343, 432, 704
    SECLVL 58
    VERIFY 63, 465
  start options affecting 553
SESSLIM operand 59
SHM operand
  PATH definition statement
    switched major node 457
  VTAM restrictions on NCP definition statements 378
SHMTIM operand
  PATH definition statement (switched major node) 457
sift-down effect 13
sifting 13
SIRFMSG start option 625
SIZE operand, VTAM internal trace 650
SLOWPT buffer pool start option operand 568
SLOWVAL start option 625
SLUALMSG start option 626
SLUINIT operand, NETSRVR definition statement 532
SMEAUTH start option 626
SMS option, VTAM internal trace 649
SNAPREQ start option 627
SNETMASK operand
  NCP major node 307
  switched major node 398, 400
SNVC operand
  NEXTCP definition statement 512
SOC (serial optical channel) line 372
SONLIM start option 627
SONSCIP operand (APPL definition statement) 59
SORDER start option 628
SPAN operand
  APPL definition statement 60
  CDRM definition statement 129
  CDRSC definition statement 111
  GROUP definition statement
    channel-attached NCP 79
    channel-to-channel 79
    external communication adapter peripheral node 180
    external communication adapter subarea node 180
    multipath channel (MPC) 79
  LINE definition statement
    channel-attached NCP 79
    channel-to-channel 79
    external communication adapter peripheral node 180
    external communication adapter subarea node 180
    multipath channel (MPC) 79
  LOCAL definition statement 200
  LU definition statement
    local SNA major node 234
    model major node 287
    switched major node 458
  PU definition statement
    channel-attached NCP 79
    channel-to-channel 79
    external communication adapter peripheral node 180
    external communication adapter subarea node 180
    local SNA major node 234
    model major node 287
    multipath channel (MPC) 79
    switched major node 458
SRBEXIT operand
  APPL definition statement 60
SRCHRED start option 629
SRCOUNT operand (CDRSC definition statement) 111
SRCOUNT start option 629
SRCVPAC operand (MODEENT macroinstruction) 712
SRQRETRY 181
SRQTIME 181
SRTIMER operand (CDRSC definition statement) 112
SRTIMER start option 630

SSCP option, VTAM internal trace 649
SSCPDYN start option 630
SSCPFM operand
  APPL definition statement 61
  LOCAL definition statement 201
  LU definition statement
    local SNA major node 234
    LU group major node 252
    model major node 288
    switched major node 458
  VTAM restrictions on NCP definition statements 378
SSCPID start option 631
SSCPNAME start option 631
SSCPORD start option 632
SSDTMOUT start option 633
SSEARCH start option 633
SSNDPAC operand (MODEENT macroinstruction) 712
start options
  ALSREQ 560
  APPNCOS 561
  ASIRFMSG 561
  ASYDE 562
  AUTHLEN 562
  AUTORTRY 562
  AUTOTI 563
  BN 564
  BNDYN 564
  BNORD 565
  BSCMDRS 565
  BSCTMOUT 566
  buffer pools 566
  CACHETI 572
  CDRDYN 572
  CDRSCTI 573
  CDSERVR 574
  CDSREFER 574
  CINDXSIZ 574
  CMPMIPS 575
  CNMTAB 576
  coding format 533
  COLD or WARM 576
  CONFIG 576
  CONNTYPE 577
  CPCDRSC 578
  CPCP 578
  CSA24 580
  CSALIMIT 579
  DATEFORM 580
  DIRSIZE 581
  DIRTIME 582
  DISCNTIM 582
  DLRTCB 583
  DSPLYDEF 584
  DSPLYMAX 584
  DSPLYWLD 585
  DYNADJCP 586
  DYNASSCP 587
  DYNDLGMD 587
  DYNHPPFX 588
  DYNLU 588
  DYNMODTB 589
start options *(continued)*
  DYNPUPFX 589
  DYNVNPFX 589
  ENCRPREF 590
  ENCRYPTN 590
  ENHADDR 591
  ESIRFMSG 592
  FLDTAB 593
  FSIRFMSG 593
  GWSSCP 594
  HNTSIZE 594
  HOSTPU 595
  HOSTSA 595
  HOTIOTRM 596
  HPR 597
  HPRPST 598
  HSRTSIZE 599
  INITDB 600
  INOPDUMP 600
  IOINT 600
  IOMSGLIM 601
  IOPURGE 601
  IRNSTRGE 602
  ISTCOSDF 603
  LIMINTCP 603
  LIST 604
  LISTBKUP 604
  MAINTLVL 605
  MAXLOCAT 605
  MAXLURU 606
  MAXSSCPS 606
  MAXSUBA 606
  MIHTMOUT 607
  MSGLEVEL 607
  MSGMOD 608
  MXSAWBUF 609
  MXSSCPRU 609
  MXSUBNUM 609
  NCPBUFSZ 610
  NETID 610
  NMVTLOG 610
  NODELST 612
  NODETYPE 612
  NOTNSTAT 637
  NQNMODE 613
  NSRTSIZE 613
  NUMTREES 614
  OSIEVENT 614
  OSIMGMT 615
  OSRTSIZE 616
  PDTRCBUF 617
  PIUMAXDS 617
  PLUALMSG 617
  PPOLOG 618
  PROMPT or NOPROMPT 618
  PSRETRY 619
  PSSTRACE 619
  RESUSAGE 620
  ROUTERES 621
  SAWMAXDS 623
  SAWMXQPK 623

start options *(continued)*
SDLCMDRS 623
SECLVLCP 624
SIRFMSG 625
SLUALMSG 626
SMEAUTH 626
SNAPREQ 627
SONLIM 627
SORDER 628
SRCHRED 629
SRCOUNT 629
SSCPDYN 630
SSCPID 631
SSCPNAME 631
SSCPORD 632
SSDTMOUT 633
SSEARCH 633
STRGR 633, 634
STRMNPS 634
SUPP 635
SWNORDER 636
TNSTAT 637
TRACE 638, 642
TRANSLAT 651
UPDDELAY 652
USSTAB 653
VERIFYCP 653
VFYRED 654
VFYREDTI 654
VRTG 655
VRTGCPCP 656
VTAMEAS 656
XCFINIT 657
XNETALS 657
START-STOP operand (KEEPMEM macroinstruction) 729
STORAGE operand 477
STRGR start option 633, 634
STRMNPS start option 634
subarea mapping table 723
subarea node, compared to APPN nodes 596
SUBAREA operand
BUILD definition statement 379
CDRM definition statement (CDRM major node) 130
GWPATH definition statement (CDRM major node) 130
HOST definition statement 379
NETWORK definition statement 380
PCCU definition statement 379
PU definition statement
external communication adapter subarea node 181
switched major node 459
subarea-to-APPN class-of-service mapping table, defining 524
SUBPORTS operand (FRSESET definition statement) 380
SUBSTUT operand
COS macroinstruction 667
SUPP operand (USSMSG macroinstruction)
terminal operator messages 750
VTAM operator messages 750
SUPP start option 635
switched major node, defining 392
SWNORDER operand
GROUP and LINE definition statements
external communication adapter major node 182
NCP major node 380
SWNORDER start option 636
SYNCLVL operand (APPL definition statement) 61
syntax checker (Definition Source Validation) 8
syntax diagrams, how to read 15
syntax notation
definition statements 9
macroinstructions for tables and filter 10
SYS1.ASAMPLIB
logon mode table 698
USS table 732
SYS1.VTAMLIB
installing COS table 666
installing interpret table 693
installing logon mode table 700
installing USS table 735
SYSCNTRL definition statement 298

T

T1TIMER operand
NCP major node 307
switched major node 398, 400
T2TIMER operand
NCP major node 307
switched major node 398, 400
table
associated LU 661
class-of-service 664
interpret 690
logon mode 698
message-flooding prevention 714
model name 720
USS 731
TABLE operand (USSTAB macroinstruction) 751
takeover restrictions
ANS operand (PU definition statement in switched major node) 262, 411
ANS operand on NCP definition statements 321
dynamic restrictions 496
TCP/IP major node 3
TCPNAME start option
start option definition 636
syntax diagram 534
TERM operand
LOCAL definition statement 201
LU definition statement
model major node 289
switched major node 459
VTAM restrictions on NCP definition statements 381
TERMINAL definition statement
VTAM restrictions on NCP definition statements 301
terminal operator commands 757
TEXT operand (USSMSG macroinstruction) 751

TGCONF operand 459
TGN operand
PCCU definition statement 381
PU definition statement
channel-to-channel 80
channel-to-NCP 80
external communication adapter subarea node 183
local SNA major node, for APPN connections 234
multipath channel (MPC) 80
NCP major node 381
switched major node 460
TGP (APPN transmission group profile)
defining 513
described 5
IBM-supplied transmission group profiles 824
IBMTGPS (member list containing the IBM-supplied TGP) 513
TGP definition statement 513
TGP operand
CDRM definition statement 130
PORT definition statement
XCA major node 183
PU definition statement
local SNA major node 235
model major node 289
NCP major node 382
switched major node 461
TIME operand (FLDENT macroinstruction) 719
TIMEDLM operand (USSTAB macroinstruction, USS table) 752
TIMER operand (external communication adapter subarea node) 184
TNSTAT start option 637
TO operand
ADD definition statement (dynamic reconfiguration) 500
MOVE definition statement (dynamic reconfiguration) 500
TOPO operand
PU definition statement
local SNA major node 184, 235
model major node 289
NCP major node 382
switched major node 461
trace
start options
description 638
NOTRACE (cancel trace) 641
TYPE=BUF (VTAM buffer trace) 642
TYPE=IO (VTAM I/O trace) 642
TYPE=LINE (NCP line trace) 642
TYPE=SIT (scanner interface trace) 642
TYPE=SMS (VTAM buffer request trace) 642
TYPE=VTAM (VTAM internal trace) 642
VTAM internal trace
ALL option 646
API option (applications program interface) 648
APPC option (LU 6.2 communication) 648
CIO option (channel I/O) 648
trace *(continued)*
VTAM internal trace *(continued)*
CMIP option (Common Management Information Protocol) 648
default 648
ESC option (execution sequence control) 648
HPR option (high performance routing) 648
LCS option (LAN Channel Station) 649
LOCK option (locking) 649
MODE operand 645
MSG option (messages) 649
NONE option (no traces) 649
NOTRACE operand 646
NRM option (network resource management) 649
OPTION operand 646
PIU option (path information units) 649
PSS option (process scheduling services) 649
SIZE operand 650
SMS option (storage management services) 649
SSCP option (system services control point) 649
VCNS option (VTAM Common Network Services) 649
TRACE start option 638
TRANSLAT start option 651
translation table 732
transmission group profile (APPN)
defining 513
described 5
IBM-supplied transmission group profiles 824
IBMTGPS (member list containing the IBM-supplied TGP) 513
transmission priority indicator number 665
transport resource list major node
defining 469
TRLE definition statement 470
TRLE operand 289
PU definition statement 236
TSO/VTAM, defining application programs 29
TSPROF operand (MODEENT macroinstruction) 713
TYPE operand
LOGOFF command 764
MODEENT macroinstruction 713

U

unformatted system services (USS) commands
terminal operator 757
VTAM operator 756
unformatted system services (USS) messages
coding 736
redefining 753
rules for creating 754
UPARM1 operand
CDRM definition statement 131
LINEROW definition statement
APPN class of service 676
PORT definition statement
XCA major node 185
PU definition statement
local SNA major node 236

UPARM1 operand *(continued)*
model major node 291
NCP major node 383
switched major node 462
TGP definition statement
transmission group profile 520
UPARM2 operand
CDRM definition statement 131
LINEROW definition statement
APPN class of service 676
PORT definition statement
XCA major node 185
PU definition statement
local SNA major node 237
model major node 291
NCP major node 383
switched major node 463
TGP definition statement
transmission group profile 521
UPARM3 operand
CDRM definition statement 131
LINEROW definition statement
APPN class of service 677
PORT definition statement
XCA major node 185
PU definition statement
local SNA major node 237
model major node 292
NCP major node 383
switched major node 463
TGP definition statement
transmission group profile 521
UPDDELAY start option 652
USE operand
PATH definition statement
switched major node 463
VTAM restrictions on VTAM definition statements 384
USERVAR
LOGCHAR macroinstruction 691
LOGOFF command 763
LOGON command 759
USS (unformatted system services) commands
terminal operator 757
VTAM operator 756
USS (unformatted system services) messages
coding 736
redefining 753
rules for creating 754
USS (unformatted system services) table
creating and modifying 736
installing 735
operator messages and commands 756
USS command syntax rules 757
USS table
command syntax rules 757
default USS table
operation-level 785
session-level 784
operation-level 732
order of use 733
USS table *(continued)*
session-level 731
USS command conversion examples 833
USSCMD macroinstruction 736
USSEND macroinstruction 737
USSMSG macroinstruction
for terminal operator messages 737
for VTAM operator messages 737
USSPARM macroinstruction 736
USSTAB macroinstruction 736
USSTAB operand
APPL definition statement 62
LOCAL definition statement 201
LU definition statement
local SNA major node 237
LU group major node 253
model major node 292
switched node 464
VTAM restrictions on NCP definition statements 385
USSTAB start option 653

V

VALUE operand (USSPARM macroinstruction) 752
VARTEXT operand (FLDENT macroinstruction) 719
VBUILD definition statement
adjacent control point major node 20
adjacent SSCP table 501
application program major node 29
channel-attachment major node 70
cross-domain resource major node 93
cross-domain resource manager major node 115
dynamic reconfiguration 494
local SNA major node 204
LU group major node 240
model major node 255
network node server list 530
switched major node 399
VCNS operand (APPL definition statement) 62
VCNS option, VTAM internal trace 649
VERALSID operand
PU definition statement
local SNA major node 186, 237
model major node 292
NCP major node 385
switched major node 464
VERID operand
PATH definition statement (switched major node) 465
VERIFY operand
APPL definition statement 63
PATH definition statement 465
VERIFYCP start option 653
VFYC operand (PCCU definition statement) 386
VFYLM operand (PCCU definition statement) 386
VFYOWNER operand (CDRSC definition statement) 112
VFYRED start option 654
VFYREDTI start option 654
VN operand
ADJCP definition statement 24

VNGROUP operand
GROUP definition statement
NCP major node 387
LINE definition statement
NCP major node 387
PORT definition statement
XCA major node 187
VNNAME operand
GROUP definition statement
NCP major node 387
LINE definition statement
NCP major node 387
PORT definition statement
XCA major node 187
VPACING operand
APPL definition statement 64
CDRM definition statement (CDRM major node) 132
CDRSC definition statement 113
LU definition statement
local SNA major node 238
LU group major node 253
model major node 293
switched major node 466
VTAM restrictions on NCP definition statements 387
VPATH definition statement (dynamic path update) 490
VR operand
COS macroinstruction 667
NCP dynamic path update 489
VTAM dynamic path update 493
VTAM PATH definition statement 483
VRPWS operand
NCP dynamic path update 489
VTAM dynamic path update 493
VTAM PATH definition statement 484
VRTG operand
CDRM definition statement 132
VRTG start option 655
VRTGCPCP operand
CDRM definition statement 133
VRTGCPCP start option 656
VTAM
associated LU table 661
class-of-service table 664
definition statement coding conventions 9
description of nodes 2
description of tables 6
interpret table 690
logon mode table 698
macroinstruction coding formats 10
message-flooding prevention table 714
model name table 720
names, restrictions on 14
nodes, overview 2
routes, defining 481
tables, overview 6
USS table 731
VTAM internal trace
MODE operand 645
NOTRACE operand 646
VTAM internal trace *(continued)*
OPTION operand
ALL 646
API (application program interface) 648
APPC (LU 6.2 communication) 648
CIO (channel I/O) 648
CMIP (Common Management Information Protocol) 648
default 648
ESC (execution sequence control) 648
HPR (high performance routing) 648
LCS (LAN Channel Station) 649
LOCK (locking) 649
MSG (messages) 649
NONE (no traces) 649
NRM (network resource management) 649
PIU (path information units) 649
PSS (process scheduling services) 649
SMS (storage management services) 649
SSCP (system services control point) 649
VCNS (VTAM Common Network Services) 649
SIZE operand 650
VTAM major nodes, overview of 2
VTAM-only operands on NCP definition statements, coding 300
VTAM restrictions on NCP definition statements
BUILD definition statement 297
GWNAU definition statement 299
HOST definition statement 297
LUDRPOOL definition statement 298
NETWORK definition statement 298
PUDRPOOL definition statement 298
SYSCNTRL definition statement 298
VTAMEAS start option 656
VTAMFRR operand (APPL definition statement) 64
VTAMTOPO operand
Model major node 293
NCP major node 389
Switched major node 466
XCA major node 188

W

WEIGHT operand (APPN class of service)
LINEROW definition statement 678
NODEROW definition statement 677
WRITE operand 80
LINE definition statement (channel-attachment major node) 80
TRLE definition statement 478
WTO/WTOR macroinstruction 742

X

XCA major node, defining 133
XCFINIT start option 657
XID operand
PU definition statement
local SNA major node 190, 238
VTAM restrictions on NCP definition statements 391
XNETALS operand
PU definition statement 239, 294, 468

XNETALS start option 657
xpanlim (buffer pool start option) 569
xpanno (buffer pool start option) 568
xpanpt (buffer pool start option) 569
XRF (extended recovery facility)
ANS operand
model major node 262
NCP major node 321
switched major node 411
HAVAIL operand (application program major node) 47

Readers' Comments — We'd Like to Hear from You

OS/390 SecureWay Communications Server
SNA Resource Definition Reference
Version 2 Release 8

Publication No. SC31-8565-03

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments — We'd Like to Hear from You
SC31-8565-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department CGMD / Bldg 500
P.O. Box 12195
Research Triangle Park, NC
27709-9990

Fold and Tape | **Please do not staple** | Fold and Tape

SC31-8565-03

Cut or Fold
Along Line

File Number: S390–50
Program Number: 5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC31-8565-03