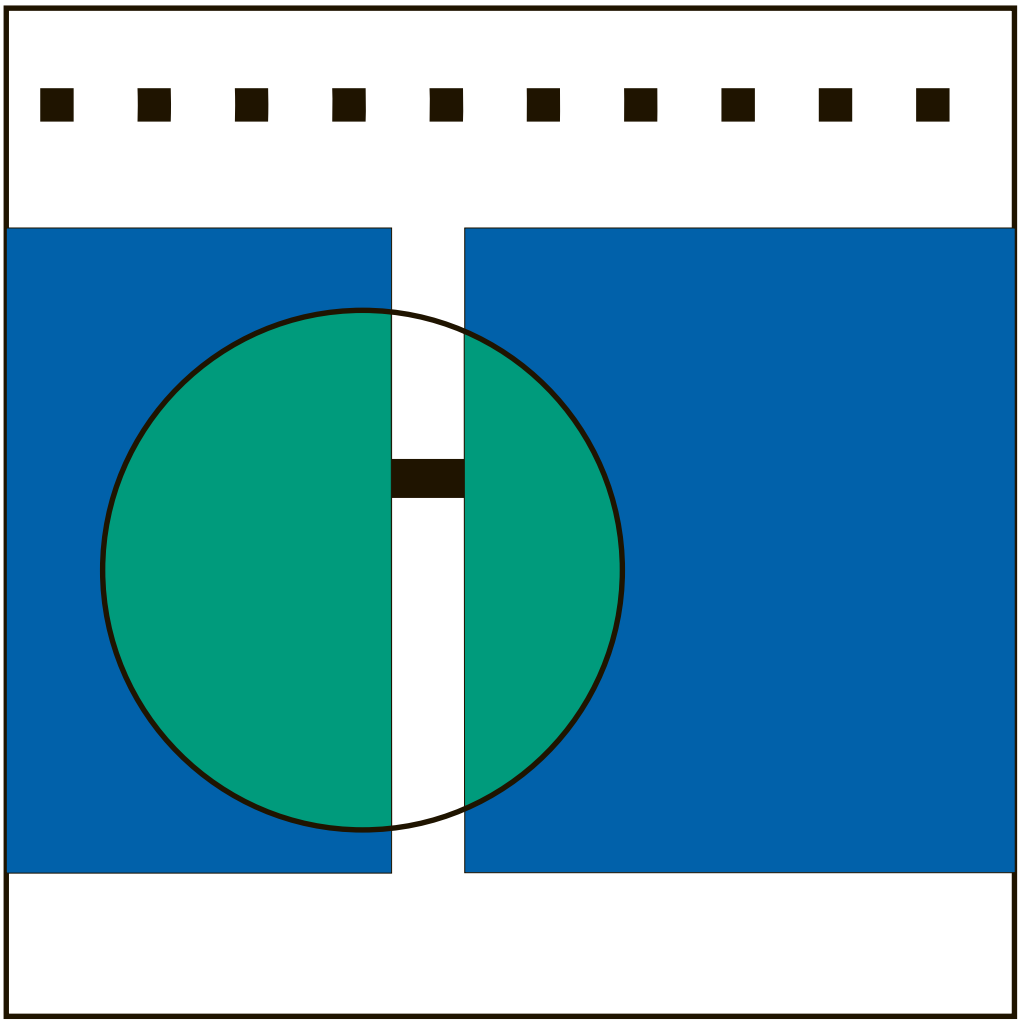
VTAM™

SC31-6498-01

Resource Definition Reference

Version 4 Release 2 for MVS/ESA, VM/ESA, and VSE/ESA



VTAM™

SC31-6498-01

Resource Definition Reference

Version 4 Release 2 for MVS/ESA, VM/ESA, and VSE/ESA

Note!

Before using this document, read the general information under “Notices” on page vii.

Second Edition (February 1995)

This edition applies to the Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM), an IBM licensed program:

VTAM Version 4 Release 2 for MVS/ESA (program number 5695-117)
VTAM Version 4 Release 2 for VM/ESA (program number 5654-010)
VTAM Version 4 Release 2 for VSE/ESA (program number 5686-065).

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 E15
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. 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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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
USA

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms, denoted by an asterisk (*) at their first occurrence in this publication, are trademarks of the IBM corporation in this country or other countries or both:

ACF/VTAM
Advanced Peer-to-Peer Networking
AIX
AnyNet
Application Development
APPN
AS/400
AT
Common User Access
CUA
DRDA
ESCON
ES/9000
ES/9370
ESA/370
IBM
MVS/ESA
MVS/XA
NETCENTER
NetView
NTuneMon
NTuneNCP
OS/400
RACF
SAA
System/370
Virtual Machine/Enterprise System
Virtual Machine/Extended Architecture
VM/ESA
VM/XA
VSE/ESA
VTAM

About This Book

This book helps system programmers define resources to VTAM*. 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.

VM,VSE Use this book in conjunction with the *VTAM Overview for VM/ESA and VSE/ESA* flyer included with this book.

To plan for the tasks of installing VTAM and defining resources to it, consult *Planning for NetView, NCP, and VTAM*.

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 *VTAM Network Implementation Guide*.

This book contains the specifics of resource definition.

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 to define routes, adjacent SSCP tables, and dynamic reconfiguration definitions.

- 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, “VM/SNA Console Support (VSCS)”

  The use of the DTIGEN macroinstruction and instructions on coding its operands are covered in this chapter. Coding considerations specific to VSCS users are also described.

- Chapter 6, “User-Defined Tables and Data Filter”

  This chapter describes how to create or modify logon mode tables, associated LU tables, model name tables, class-of-service tables, interpret tables, unformatted system services (USS) tables, and message-flooding prevention table. It also describes how to code the SAW data filter.

- Appendix A, “IBM-Supplied Tables”

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.

Terms Used in the VTAM Library

Clarification of some terms used in the VTAM library follows:

- The term *Advanced Peer-to-Peer Networking* (APPN*)* represents VTAM's implementation of APPN.
- The terms *CPCP* and *CP-CP* are similar but have different meanings. *CPCP* is the name of a start option or operand. *CP-CP* refers to CP-CP sessions between control points.
- The term *end node* represents an APPN end node.
- The term *integrated communication adapter (ICA)* is used to represent all of the following, except where it is necessary to use the specific term:
  - Communication adapter
  - Telecommunications Subsystem Controller
  - IBM Token-Ring Subsystem Controller
  - Workstation Subsystem Controller
  - Multi-Protocol Communication Subsystem Controller.
- The term *network node* represents an APPN network node.
- The term *VTAM* refers to the VTAM program for V4R2 unless otherwise stated.

Symbols Used in the VTAM Library

The following symbols are used in the VTAM library to indicate information that applies only to a particular operating system or processor:

MVS Indicates information that applies only to MVS/ESA*.

VSE Indicates information that applies only to VSE/ESA*.

VM Indicates information that applies only to VM/ESA*.

These symbols either precede or follow unique information in the VTAM library. Following is an example of how these symbols are used:

VM VSCS is a VTAM application program and must be defined to VTAM.

What Is New in This Book

Information has been added to this book to reflect the new functions in VTAM V4R2 for VM and VSE. Many of the functions were introduced for MVS in V4R1 and V4R2. Changes have also been made as a result of user comments.

The glossary has been deleted from this publication. For definitions of the terms and abbreviations used in this book, refer to the *VTAM Glossary*.

This book was previously published:

- **MVS** May, 1994 with order number SC31-6498-00
- **VM,VSE** March, 1992 with order number SC31-6438-1

The following sections contain summaries of the changes that have been made for each new function.

Note: A VM or VSE icon appearing in the following sections denotes a function that is new only to that operating system. For instance, the VERIFYCP start option is marked with a VM icon, as it is new to VM and does not apply to VSE (but has applied to MVS since V4R1).

Network Dynamics

The following functions enable dynamic access to a network. These functions can eliminate or reduce the predefinition required for resources in your network. In addition, these functions provide increased end user availability and automatic backup.

Advanced Peer-to-Peer Networking (APPN) Support

VTAM now supports Advanced Peer-to-Peer Networking. With VTAM's support for APPN, the range of system platforms that support APPN now includes System/370*, System/390*, and ES/9000* mainframe processors.

With this support, VTAM nodes and composite network nodes can be dynamic participants in an APPN network, allowing dynamic access to mainframe resources from other APPN nodes and dynamic use of the SNA backbone network as a transport between APPN nodes.

Switched and nonswitched type 2.1 nodes can be connected as APPN nodes in an X.25 network or in a channel-attachment major node for SDLC integrated communication adapter (ICA) support. Switched type 2.1 nodes can be connected as APPN nodes in a token-ring LAN environment.

Included are end node, network node, and central directory server support. Also included are border node, host-to-host channel, connection network, dependent LU server and virtual-route-based transmission group support.

VTAM V4R2 operating as a network node or end node in an APPN network continues to support subarea networks as well, including support for dependent LUs (LU0, LU1, LU2), mainframe-based application programs, and application programming interfaces.

Use of composite network node boundary function requires, at a minimum, NCP V6R2.

Multiple Network Connectivity: APPN multiple network connectivity (also known as border node) provides:

- Communication between networks with different network IDs through APPN. Although APPN protocols are allowed between different networks, topology information is not shared. APPN searches can extend over more than one network.
- Division, or clustering, of an APPN network into two or more subnetworks with the same network ID. Clustering might be most desirable when you have a large APPN network and want to shield smaller subnetworks from the large volume of topology information and unnecessary directory search requests.

A network node with APPN multiple network connectivity, also known as a border node, can maintain CP-CP connectivity with a network node with a different NETID. APPN topology information does not cross the border node connection or APPN subnetwork boundary. An APPN subnetwork boundary is assumed by VTAM when a border node is connected to a node with a different network identifier or can be explicitly defined through system definition.

Border nodes can be used to set up subnetwork boundaries between nodes with the same network identifier. The subnetwork boundary is defined by the presence of an APPN subnetwork link, which is created through system definition.

Use of border node support in configurations where NCP provides the boundary function for the APPN connections between two APPN subnetworks requires, at a minimum, NCP V7R1.

Host-to-Host Channel Support: The APPN host-to-host channel function allows a VTAM node to attach to a second VTAM node with a multipath channel connection using APPN protocols. The function enables two adjacent APPN host nodes to be connected by a channel interface.

Use of host-to-host channel support requires, at a minimum:

- IBM 3088 Multisystem Channel Communication Unit, or equivalent; for example, the IBM 3172 Interconnect Controller
- IBM Enterprise System Connection (ESCON)* channel
- Channel-to-channel adapter (CTCA).

Connection Network Support: A connection network is a representation within an APPN network of a shared-access transport facility (SATF), such as a local area network (LAN), that allows attached APPN nodes to:

- Communicate without having individually defined connections to one another
- Establish sessions without routing through a network node server.

The connection across the LAN is defined as a connection to a virtual node. By defining a single virtual node, each node is adjacent to every other node attached to the LAN.

Note: A VTAM V4R1 network node can calculate a route across a connection network, but it cannot be a session endpoint on such a route.

Supported SATFs for this function include:

- Token-ring LAN
- Fiber distributed data interface (FDDI)
- Carrier sense multiple access with collision detection (CSMA/CD).

Connection network support is provided for VTAM through:

- An NCP (through NCP/Token-Ring Interconnection [NTRI]). NCP V7R1 is required for NTRI support of the connection network function.
- An IBM 3172 Interconnect Controller (through an external communication adapter)
- A token-ring (through an ICA). Connection network support for ICA connections requires no additional products.

APPN adds the following to this book:

- A new adjacent control point (ADJCP) major node
- A new transport resource list (TRL) major node
- The following new definition statements:
  - For APPN class of service:
    - APPNCOS
    - LINEROW
    - NETSRVR
    - NODEROW
  - For transmission group profile:
    - TGP
- A new CPNAME operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - Local SNA
  - Packet
  - NCP
- The following new operands on the GROUP, LINE, and PU definition statements in the NCP, channel-attachment (SDLC), and packet major nodes and on the PU definition statement in the switched, local SNA, and model major nodes:
  - CAPACITY
  - CONNTYPE
  - COSTBYTE
  - COSTTIME
  - CPCP
  - DYNADJCP
  - NN
  - PDELAY
  - SECURITY VM
  - TGP
  - UPARM1
  - UPARM2
  - UPARM3
  - VERALSID
- A new TOPO operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - Local SNA
  - Model
  - NCP
  - Packet
  - Switched
- A new TRLE operand on the PU definition statement in the local SNA major node
- A new NATIVE operand on the PU definition statement in the following major nodes:

  - Channel-attachment (SDLC)
  - LAN
  - Local SNA
  - Model
  - NCP
  - Packet
  - Switched
- The following new operands on the GROUP definition statement in the NCP, external communication adapter (XCA), channel-attachment (SDLC), LAN, and packet major nodes:
  - DYNPU
  - DYNPUPFX
- A new DISJOINT operand on the CDRM definition statement in the cross-domain resource manager major node
- A new GROUP definition statement in the cross-domain resource manager major node
- A new DWACT operand on the PU definition statement in the switched major node
- The following new operands on the CDRSC definition statement in the cross-domain resource major node:
  - CPNAME
  - NETSRVR
  - REGISTER
  - SRCOUNT
  - SRTIMER
- The following new operands on the LINE definition statement in the NCP major node:
  - VNGROUP
  - VNNAME
- The following new operands on the PORT definition statement in the external communication adapter (XCA) major node:
  - TGP
  - VNGROUP
  - VNNAME
- The following new operands on the PORT definition statement in the LAN (peripheral) major node:
  - CAPACITY
  - COSTBYTE
  - COSTTIME
  - PDELAY
  - SECURITY
  - TGP
  - UPARM1
  - UPARM2
  - UPARM3
  - VNGROUP
  - VNNAME

- A new REGISTER operand on:
  - The APPL definition statement in the application program major node
  - The CDRSC definition statement in the cross-domain resource major node
  - The LU definition statement in the channel-attachment (SDLC), local SNA, LU group, model, packet, NCP, and switched major nodes
  - The LOCAL definition statement in local non-SNA major nodes
- Changes to the DYNLU operand on the GROUP, LINE, and PU definition statements in the following major nodes:
  - Channel-attachment (SDLC)
  - NCP
  - Packet

  and on the PU definition statement in the following major nodes:
  - Local SNA
  - Model
  - Switched
- Changes to the TGN operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - Local SNA
  - Model
  - NCP
  - Packet
  - Switched
- Changes to the XID operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - Local SNA
  - Model
  - NCP
  - Packet
- The following new start options:
  - APPNCOS
  - BN
  - BNDYN
  - BNORD
  - CACHETI
  - CDSERVR
  - CONNTYPE
  - CPCP
  - DIRSIZE
  - DIRTIME
  - DYNADJCP
  - INITDB
  - NODETYPE
  - NUMTREES
  - RESUSAGE
  - ROUTERES

  - SECLVLCP
  - SNVC
  - SORDER
  - SRCHRED
  - SRCOUNT
  - SRTIMER
  - SSEARCH
  - VERIFYCP VM
- Changes to the following start options:
  - CDRSCTI
  - DYNLU
  - HOSTSA
  - XNETALS
- A new IDTYPE operand on the TRACE start option
- The following new IBM-supplied definitions:
  - Adjacent cluster
  - Border node class-of-service (COS) mapping
- A new APPNCOS operand on the MODEENT macroinstruction in the IBM-supplied default logon mode table, ISTINCLM
- Samples of the following new IBM-supplied APPN class-of-service (COS) definitions:
  - CPSVCMG
  - SNASVCMG
  - #BATCH
  - #BATCHSC
  - #INTER
  - #INTERSC
  - #CONNECT
- Samples of the following updated IBM-supplied default tables:
  - ISTINCDT (session-level USS table)
  - ISTINCLM (logon mode table)
  - ISTINCNO (operation-level USS table).

Definition of Spare SDLC Lines

In pre-V4R2 releases, line definitions in NCP are statically defined and the NCP must be regenerated to add new lines into an NCP definition.

In VTAM V4R2, you can predefine lines so that a level of dynamics is available for NCP line definitions when using VTAM, NCP, NTuneMon*, and NTuneNCP*.

You can now define a spare line in your NCP definition for physical line slots that have no physical lines attached. The spare line acts as a place holder for a physical line to be attached later.

This function also allows you to specify that a line attached to the line slot and in use can be redefined later as a spare line.

Definition of spare SDLC lines requires, at a minimum:

- NCP V7R2

- NTuneMon
- NTuneNCP.

See the NCP major node section for information about the following new values on the USE operand:

- REDEF
- SPARE.

Dynamic Reconfiguration of PUs on a Frame Relay Line

This function enables you to dynamically add and delete type 1 PUs to and from an NCP frame relay line. It also allows you to dynamically add and delete NCP FRSESET definition statements.

Use of dynamic reconfiguration of PUs on a frame relay line requires, at a minimum, NCP V6R2.

See the NCP major node section for information about the NCP definition statements that are validated by VTAM in association with dynamic reconfiguration for NCP frame relay resources.

Dynamic Reconfiguration of Retry Values and 3174 Group Addresses on a Nonswitched SDLC Line

For a PU that is to be dynamically reconfigured, you can now specify the time interval between retry sequences and the number of retry sequences for error recovery attempts made when problems occur during transmission over a link. You can also specify 3174 group addresses for PUs that are attached to peripheral nodes on a nonswitched SDLC link and that are to be dynamically reconfigured.

Use of these dynamic reconfiguration capabilities requires, at a minimum, NCP V6R2.

See the NCP major node section for information about:

- A new GP3174 operand on the PU definition statement in the NCP major node
- Changes to the RETRIES operand on the PU definition statement in the NCP major node.

Enabling Dynamic Definition of Cross-Domain Resources with the CDRDYN Start Option

You can now use a new CDRDYN start option to specify whether a host is authorized to dynamically define cross-domain resource representations of cross-domain cross-network subarea or APPN resources when a session request is received from or sent to an adjacent SSCP or CP. In pre-V4R2 releases, this capability is available in APPN hosts and in subarea hosts only after the host cross-domain resource manager is activated.

See Chapter 4, “Start Options” for information about the new CDRDYN start option.

Session Enhancements for LEN CP Independent Logical Units

In pre-V4R2 releases, a dynamic independent logical unit (ILU) that has the same name as the control point associated with its adjacent link station (otherwise known as a LEN CP ILU) is not represented as a cross-domain resource until it initiates a session. Consequently, if another resource attempts to initiate a session with this ILU before this ILU initiates a session, the other resource's session request fails.

In VTAM V4R2, you can use a new start option to indicate that VTAM will allow other resources to initiate sessions with dynamic ILUs that have the same names as the control points associated with their adjacent link stations before these ILUs initiate sessions.

See Chapter 4, “Start Options” for information about a new CPCDRSC start option.

Operator Interface Enhancements

The following functions enhance VTAM's operator interface.

Abbreviations for Buffer Pool Start Options

In pre-V4R2 releases, you can use only the IBM-supplied 5- to 7-character full buffer pool names to specify buffer pool start options. Now, you can use either of two IBM-supplied abbreviated forms in addition to the traditional full buffer pool names. The shorter forms now available are the buffer pool IDs and buffer pool abbreviations used on the DISPLAY BFRUSE command in this and in pre-V4R2 releases.

See Chapter 4, “Start Options” for a description of the buffer pool start options and a list of the full and abbreviated names you can specify for the *poolname* operand.

Display of a Maximum Number of Resources

You can now limit or expand certain displays to a specific number of resources or lines of output.

See Chapter 4, “Start Options” for information about a new DSPLYMAX start option.

Display of VTAM Maintenance Level

VTAM now enables you to display the maintenance level and component identifier of the VTAM you have running in a particular host processor. You code the maintenance level on a new start option. The component identifier is automatically displayed when you issue a DISPLAY VTAMOPTS command.

See Chapter 4, “Start Options” for information about a new MAINTLVL start option.

Inactivating LUs in a Pending-Notify State

An application program that has a LOSTERM exit routine but no NSEXIT exit routine must issue a CLSDST macroinstruction when the LOSTERM exit is driven to terminate a session. Some application programs do not do this and the session becomes hung. If a VARY INACT command is issued against the LU in the hung session, the LU will hang in a pending-notify state. The only recovery is to recycle the application program to clean up one hung session or halt VTAM.

This function allows you to force the resource to an inactive state when the application program does not issue a CLSDST macroinstruction.

See the Application Program Major Node section for information about a new LOSTERM operand on the APPL definition statement.

Scheduled Automatic Reloading (VSE)

With this support, the operator can schedule the IPL of the IBM 3745 Communication Controllers. The operator can also cancel an IPL that was previously scheduled and display the time of the scheduled IPL. Refer to *Guide to Timed IPL and Rename Load Module* for complete information.

This function requires an appropriate model of the IBM 3745 Communication Controller with the corresponding engineering changes. Refer to the *VTAM Program Directory* for V4R2 for more information.

See Chapter 4, “Start Options” for information about a new DATEFORM start option.

Specifying of Replaceable Constants on Start Options

In pre-V4R2 releases, to change most of the values in the replaceable constants module, ISTRACON, you have to “zap” the ISTRACON module and then restart VTAM so that the new values take effect.

Now, the ISTRACON module is no longer part of VTAM. All replaceable constants must be set with start options.

In addition, you can reset some of the values with the MODIFY VTAMOPTS command while VTAM is running. A value you specify on the MODIFY VTAMOPTS command overrides a value specified on a start option. You can also use the DISPLAY VTAMOPTS command to display the existing values on one or more start options.

See Chapter 4, “Start Options” for information about the following new start options:

- ASIRFMSG
- BSCTMOUT
- CINDXSIZ
- ESIRFMSG
- FSIRFMSG
- HNTSIZE
- HSRTSIZE
- INOPDUMP VM
- IOMSGLIM
- IRNSTRGE
- MAXLURU
- MAXSSCPS
- MIHTMOUT
- MXSAWBUF
- MXSSCPRU
- MXSUBNUM
- OSRTSIZE
- PDTRCBUF
- PLUALMSG
- SIRFMSG
- SLUALMSG

- SNAPREQ
- SSDTMOUT
- TRANSLAT
- VTAMEAS.

Start List Processing Enhancements

VTAM now responds differently from pre-V4R2 releases when it cannot find the user-defined start option list, ATCSTR00. In pre-V4R2 releases, VTAM issues a warning message and start processing continues using internal default values.

Now, VTAM halts start processing and issues a prompting message giving you a choice of actions. You must respond to the prompting message `IST1311A ENTER START OPTION OVERRIDES OR ENTER HALT TO EXIT VTAM` for processing to continue. If you prefer not to use ATCSTR00 and you do not want to get the prompting message, code an empty ATCSTR00 start option list (containing only comments) so that start processing will not be interrupted. The prompting message gives you the following choice of actions:

- Continue start processing using internal default values.
- Specify a different start option list, ATCSTR*yy*, where *yy* specifies the identifier of the start option list you want VTAM to use in place of ATCSTR00.
- Halt VTAM, which cancels the start attempt. You can then make the ATCSTR00 start option list available and try again.

Errors found in a start option list are also now handled differently in VTAM. In pre-V4R2 releases, VTAM notifies you to correct and reenter any start options that are specified incorrectly. Now, you have several choices:

- Continue to process the start option list, and then reenter the correct start options.
- Use a different ATCSTR*yy* start option list, where *yy* specifies the identifier of the start option list you want to use. This cancels the start attempt and restarts processing with the specified start option list.
- Halt VTAM, which cancels the start attempt. You can then correct and reenter the start options and try again.

VTAM now also enables you to use a new LISTBKUP start option in a start option list. You can use the new LISTBKUP start option to specify how you want VTAM to react if it encounters an error while processing a start option list. VTAM can react by:

- Processing a different start option list, which you specify
- Resetting options to their previous values
- Setting all valid options and prompting you to correct and reenter any options that are specified incorrectly.

If you do not specify in a start option list how you want VTAM to react if it encounters an error while processing that start option list, it will prompt you to:

- Continue to process the start option list, and then reenter the correct start options.

- Use a different ATCSTR*yy* start option list, where *yy* specifies the identifier of the start option list you want to use. This cancels the start attempt and restarts processing with the specified start option list.
- Halt VTAM, which cancels the start attempt. You can then correct and reenter the start options and try again.

VSE In addition, in pre-V4R2 releases, you can start VTAM with a customized set of start options in one of the following ways:

- Changing the IBM-supplied default start option values in the ATCSTR00 start option list
- Coding a separate start option list (ATCSTRyy) and then specifying LIST=yy in response to the prompt after issuing the EXEC command or running the start procedure JCL
- Overriding IBM-supplied default start options in response to the prompt after issuing the EXEC command or running the start procedure JCL.

In VTAM V4R2, you can code a separate start option list (ATCSTRyy) and then specify that start option list name on the EXEC ISTINCVT statement in the start procedure JCL. When you issue the EXEC command or run the start procedure JCL, the ATCSTRyy start option list will be processed automatically, without you being prompted.

See Chapter 4, “Start Options” for information about the new LISTBKUP start option.

User-Defined Message-Flooding Prevention Table

The message-flooding prevention facility, shipped under object-code-only (OCO) protection since VTAM V3R1, 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 V4R2 provides a message-flooding prevention table. The table identifies a list of messages that are candidates for flooding suppression. For each candidate message, the table identifies the criteria that must be met before VTAM suppresses duplicate messages. The suppression criteria include:

- The amount of time between the original and subsequent message
- An indication of which variable text fields are to be compared
- Whether a message is suppressed from the system hardcopy log.

If the message is reissued within the specified time interval and the variable text fields contain the same information, VTAM suppresses the message.

Note: For groups of messages, suppression is based on the first message in the group.

See Chapter 4, “Start Options” for information about a new FLDTAB start option, and for information about how to code the message flooding prevention table.

USS Message Text Enhancements

You can now use several new character strings in the text of messages you define in a USS table. These character strings enable you to include the following information in USS messages, in addition to LU name:

- Date
- Network identifier
- Network-qualified name
- Time.

See the Unformatted System Services (USS) section for information about:

- The following new character strings, which can be included in the message text defined in the buffer identified by the BUFFER operand on the USSMSG macroinstruction:
  - @@@@DATE
  - @@@NETID
  - @@@@@@@@ @@@@@@@NQN
  - @@@@TIME
- The following new operands on the USSTAB macroinstruction:
  - DATEDLM
  - DATEFRM
  - TIMEDLM.

Wildcard Resource Names

Wildcard values enable an operator to expand a display by substituting special symbols to represent unspecified characters in the name of a resource. You can also use wildcard values to search for an ID when you know only part of its value. You can use the following symbols as part of the ID operand value:

Asterisk (*)	Represents a string of unspecified characters
Question mark (?)	Represents a single unspecified character.

Wildcards can be placed anywhere in the value of the ID operand. Following are several examples of how wildcard names are used:

ID=*	Displays all resource names
ID=APP*	Displays resource names that start with *APP*
ID=APPL*01	Displays resource names that begin with *APPL* and end with *01*
ID=A*P?1	Displays resource names that start with *A* and end with *Px1*, where *x* is any single character
ID=??*	Displays resource names of two or more characters
ID=(AP12,APP*,*01)	Displays the first resource, then all resource names that start with *APP*, then all resources names that end with *01*.

Wildcard support is also available for network-qualified names on the DISPLAY TRACES command.

See the Application Program Major Node section for information about:

- A new DSPLYWLD operand on the following definition statements in the application major node:

  - APPL
  - GROUP.

See Chapter 4, “Start Options” for information about a new DSPLYWLD start option.

Performance Enhancements

The following functions provide enhancements to VTAM performance.

Authorized Transmission Priority for LEN Connections

In pre-V4R2 releases, when VTAM forwards a BIND from a LEN-attached node into a subarea network, it sets the transmission priority field to medium priority (B'01') regardless of the value specified on a BIND received from a LEN-attached node. With this function enabled, VTAM now copies a transmission priority field from the received BIND onto the BIND sent into the network.

This function provides an enhancement for the processing of the transmission priority field on BFINIT. This enables a PU 2.1 exit point from the network to use the same transmission priority as the PU 2.1 entry point. When this function is not enabled, VTAM uses the previous default of medium priority (B'01') on the BIND.

This function adds the following to this book:

- A new AUTHLEN operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - Local
  - Model
  - NCP
  - Packet
  - Switched
- A new AUTHLEN start option.

Automatic Logon Enhancements

Automatic logon is an existing function in VTAM that allows an LU to establish a session with an application program (also known as a controlling primary logical unit or controlling PLU) as soon as the LU is activated. With these enhancements, you now have more control over how VTAM retries automatic logon requests when the controlling PLU is initially unavailable.

By using the LOGAPPL operand on the definition statement for an LU, you can specify a controlling PLU for an automatic logon session. Whenever the SLU is activated, VTAM attempts to establish a session between the SLU and the controlling PLU. If the controlling PLU is not available, the session request fails, and the session is retried when the controlling PLU becomes available.

In pre-V4R2 releases, VTAM cannot guarantee the automatic retry for outstanding logon requests in the following circumstances:

- The route between the SLU and the controlling PLU is dependent on an active CP-CP session.
- The controlling PLU is in a nonadjacent domain or nonnative network.
- The controlling PLU is a USERVAR whose value is not known to VTAM.

- The controlling PLU issues a CLSDST PASS to an inactive application program.

Under these circumstances, the user has to redrive the session using the VARY LOGON command.

The automatic logon enhancements provide a timer mechanism that redrives all outstanding logon requests after a user-defined period of time. Furthermore, the enhancements allow retries of outstanding automatic logon requests based on the type of adjacent node activation, including adjacent CP activation.

See Chapter 4, “Start Options” for information about the following new start options:

- AUTORTRY
- AUTOTI.

Automatic Termination of Sessions Using Too Much of IO Buffer Pool

VTAM now enables you to set a threshold at which it will terminate sessions that are using too much of the IO buffer pool. You set this threshold with a new HOTIOTRM start option.

See Chapter 4, “Start Options” for information about a new HOTIOTRM start option.

Data Compression (VSE)

This facility enables VTAM to compress the data on selected LU-LU sessions when you are using application programs in a multiple-domain network. The session partners can negotiate the use of data compression through the BIND and BIND(RSP) request units (RUs). The session partners can negotiate whether to use data compression. If compression is to be used, the session partners can negotiate the type of compression.

You can implement data compression for logical units in either an extended BIND (type 2.1, 4, or 5 physical unit) or non-extended BIND environment (type 1 or 2 physical unit). You cannot compress data for sessions with LU type 4 in a non-extended BIND environment.

VTAM supports two different compression algorithms:

Run length encoding (RLE)
: Replaces strings of identical bytes with 1-byte or 2-byte sequences

Adaptive compression (similar to Lempel-Ziv)
: Uses tables that are dynamically updated to match the data being sent or received. The original data string is replaced with a set of compression codes.

Adaptive compression for outgoing data is accomplished through the use of a new hardware instruction available when VTAM is running in certain IBM ES/9000 series processors and the ES/390 processor. Adaptive compression can also provide a significant performance improvement on any supporting processor that does not have the new compression instruction.

A new CMPMIPS start option directs the new algorithm to either spend more CPU cycles for better compression effectiveness or to save more CPU cycles with some reduction in compression effectiveness. RLE compression can be turned on and off by the CMPMIPS start option.

This function uses VSE data space storage.

Hardware-assisted data compression requires, at a minimum:

- IBM ES/9000 9021 711-based Model Processor
- IBM ES/9000 9121 511-based Model Processor
- IBM ES/390 Processor.

Refer to this book for information about:

- The following new operands on the APPL definition statement in the application program major node:
  - CMPAPPLI
  - CMPAPPLO.
- A new COMPRES operand on the MODEENT macroinstruction in the logon mode table
- The following new start options:
  - CMPMIPS
  - CMPVTAM.

Delayed Activation of Logical Lines (VSE)

This function enables you to perform an orderly activation or deactivation of a physical resource and its associated logical resources. Because there is a hierarchical relationship between the physical resources (line and physical unit) and the logical lines, you can use this function to implicitly activate a logical line when the physical resources are explicitly activated.

This function creates a hierarchy between a logical line group (within an NCP major node) that has a PHYSRSC operand coded and the physical resource indicated by the PHYSRSC operand. This includes:

- NCP Token-Ring interconnect (NTRI) physical resources (line and PU) and logical lines for INN link stations
- NCP Packet Switching Interface (NPSI) physical resources (line and PU) and logical lines
- Frame relay physical resources (line and PU) and logical lines.

This function requires, at a minimum, NCP V6R1.

See the NCP Major Node section for information about the PHYSRSC operand on the GROUP definition statement in an NCP major node. For information about NCP definition statements, refer to the *NCP, SSP, and EP Resource Definition Reference*.

VTAM can now examine the ECLTYPE operand in an NCP major node. See the NCP Major Node section for more information.

Delayed Disconnection

In pre-V4R2 releases, you can specify that a dial connection disconnect after the LU-LU sessions are terminated (DISCNT=YES) or that the connection remain indefinitely (DISCNT=NO). Now, the delayed disconnection function allows an operator or system programmer to delay disconnection of switched resources to provide sufficient time for another LU-LU session to be started.

This function adds the following to this book:

- A change to the DISCNT operand on the PU definition statement in the following major nodes:
  - Channel-attachment (SDLC)
  - LAN
  - Packet
  - Switched
- A new DISCNTIM start option.

Automatic Recovery from an Outstanding Session Request

In pre-V4R2 releases, when an application program requests a session, VTAM looks for the resource by sending a CDINIT request to the first adjacent SSCP listed in its adjacent SSCP table. If the adjacent SSCP does not own the resource, VTAM sends the CDINIT request to the next adjacent SSCP listed in the table until the resource is found.

If at any time during the search, the adjacent SSCP fails to respond to the CDINIT request, VTAM might wait indefinitely for a response.

Because VTAM is waiting for a response, the session cannot be established, even though the owning adjacent SSCP might have been found in a subsequent search.

VTAM now provides the IOPURGE start option to specify a time interval during which VTAM waits for a response to a CDINIT request sent to an adjacent SSCP. If VTAM does not receive a response during the time interval you specify, it continues searching its adjacent SSCP table until the resource owner is found or the table is exhausted.

You can also use the IOPURGE start option to specify a time interval during which VTAM waits for a response to a search request generated by a direct search list (DSRLST) or an APPN broadcast search or directed search.

See Chapter 4, “Start Options” for information about the IOPURGE start option.

Multipath Channels between Host Processors

In pre-V4R2 releases, a single transmission group (TG) between hosts consists of only one connection. Although you can define several subchannels on one link (when you install and configure your operating system), VTAM requires you to make each subchannel between hosts a separate TG.

Now, this restriction is eliminated when you implement multipath channels. With multipath channels, you can define several subchannels on one TG, for host-to-host communication. These additional subchannels expand the bandwidth of one TG by increasing the number of logical paths available on the TG, and can be across the same or multiple physical connections.

Multipath channels are supported for parallel TGs. Multipath channels are not supported for use with the IBM 3172 Interconnect Controller.

This function adds the following to this book:

- A new group used to define multipath channel support
- The following new operands on the LINE definition statement in the channel-attachment (MPC) major node:
  - READ
  - WRITE.

Substitute Class-of-Service Name (VSE)

The new SUBSTUT operand on the COS macroinstruction enables VTAM to provide a substitute class-of-service (COS) entry to be used when that VTAM does not recognize the COS entry specified by the VTAM that requests the user session.

See the Class of Service (COS) section for a description of and the syntax for the SUBSTUT operand on the COS macroinstruction.

Problem Diagnosis

The following functions are enhancements to VTAM's problem determination and isolation capabilities.

Buffer Contents Trace Enhancements

In pre-V4R2 releases, the buffer contents trace captures data flowing inbound to VTAM and outbound to the network in 256-byte records. Any data beyond the 256-byte length is truncated.

Now, the buffer contents trace can record all data contained in each buffer, making it easier to diagnose problems in the network.

VM Use of the buffer contents trace enhancements requires, at a minimum, VM/ESA V1R2.2.

See Chapter 4, “Start Options” for information about a new AMOUNT operand on the TRACE start option.

IPL and Dump Contention Improvements (VSE)

This function improves the way race conditions in dumping channel-attached NCPs are handled. This function allows multiple hosts to specify AUTODMP and AUTOIPL for the same NCP. More than one host might take a dump successfully.

This function requires, at a minimum, SSP V3R7.

This function changes the AUTODMP and AUTOIPL operands on the PCCU definition statement in the NCP major node.

Scanner Interface Trace (SIT) Enhancements (VSE)

You can now reduce the volume of trace output and focus problem determination on specific connectivity subsystem (CSS) resources of the IBM 3745 Communication Controller by using scanner interface trace selectivity.

When you activate a scanner interface trace for an NCP target adapter (such as a CSS-attached ESCON* adapter, a token ring port, or a communication line adapter port), NCP can activate multiple trace points. A trace point can be activated in the target adapter and another trace point can be activated in the controller bus processor connecting the CSS and central control unit (CCU) components. You can specify that trace points be activated either at the target adapter only, the controller bus processor only, or at both trace points.

Use of these enhancements requires, at a minimum, NCP V6R2.

See Chapter 4, “Start Options” for information about a new TRACEPT value on the TRACE start option.

VIT Format Enhancements (VSE)

Data is now always automatically recorded for the following VIT options:

- API
- MSG
- PIU
- SSCP
- NRM.

Data is recorded for the above options internally or externally, depending on how the trace is started.

Note: You cannot turn off tracing for the above options.

VTAM trace records are also now more understandable and easier to read because VIT entries are split into logical pieces and labeled to indicate what the data represents. In addition, during the processing of an external trace, more temporary internal buffers are used to reduce the number of lost VSE trace records.

See Chapter 4, “Start Options” for information about changes to the following start options:

- NOTRACE
- TRACE.

Record Application Program Interface (API) Enhancements

The following functions enhance VTAM's record application program interface.

Persistent LU-LU Sessions (VSE)

Persistent LU-LU session support can be used by a VTAM application program to facilitate recovery following a failure. The support can also be used to manage the planned takeover of an application program that has enabled persistence.

When a VTAM application program fails after it has enabled persistence, VTAM retains the LU-LU sessions and associated LU-modes. For sessions to be retained, VTAM and the operating system must remain active. When a failure occurs, the application program has two ways to manage the recovery:

- Another instance of the failing application program can restart, recover the connection to VTAM, and restore sessions.
- An alternate application program that has been started already can recover the connection to VTAM and restore sessions.

Following the recovery of the application program, these sessions and modes must be restored to permit continued use.

When a planned takeover occurs, the takeover application program connects to VTAM and restores sessions, even though the original application program has not failed.

In addition, this function enables VTAM to track additional information for recovery of LU 6.1 and LU 6.2 sessions that are capable of persistence and are supported through the record application programming interface (RAPI).

For LU 6.1 sessions, VTAM copies BID data, BIS data, and the session qualifier pair. For LU 6.2 sessions, VTAM copies BID data, BIS data, FMH-5 data, the MODENAME, the session instance identifier, and the session qualifier pair.

This function uses data queueing. Inbound data is queued in a VTAM data space, rather than in the application program address space for all VSE/ESA application programs.

See Chapter 4, “Start Options” for information about new, changed, and deleted buffer pools.

System and Configuration Management

The following functions enhance your ability to set up, manage, and control the resources in your network.

Addressing Enhancement (VSE)

31-bit addressing support allows VTAM to put most of its code and control block storage in the 31-bit addressing space. This code includes VTAM private storage and system GETVIS storage. Storage below the 16Mb line can be used by application programs that use 24-bit addressing.

See Chapter 4, “Start Options” for information about the following new start options:

- SGA24
- SGALIMIT

Adjacent SSCP Lists for Cross-Domain Resources

In pre-V4R2 releases, selection of adjacent SSCPs for routing by cross-domain resources is determined by one of the following:

- Value of the CDRM operand on the CDRSC definition statement
- Predefined adjacent SSCP tables
- Network default values.

You can also write a session management exit routine to override the CDRM value defined for the cross-domain resource (CDRSC).

Now, you can also assign a list of adjacent SSCPs to a CDRSC as the route to use for cross-domain and cross-network session requests. The list defines the only route available when establishing a session with this resource. If this route is not available, the session fails. This function should be used when close control of route selection is desirable.

This function adds the following to this book:

- A new ADJLIST definition statement in the adjacent SSCP table
- A new ADJLIST operand on the following definition statements in the cross-domain resource major node:
  - CDRSC
  - GROUP.

Dependent LU Server

The dependent LU server function enables a VTAM network node or interchange node to act as a dependent LU server (DLUS) that can provide SSCP services for dependent LUs located on remote end nodes or network nodes. The end node or network node can be in the same APPN network as the DLUS or in another APPN network. The DLUS provides SSCP services through standard SSCP-PU and SSCP-LU session flows that have been encapsulated and sent over an LU 6.2 session to a dependent LU requester (DLUR). An APPN-only path using the CPSVRMGR class of service must exist between the VTAM DLUS node and the DLUR. A DLUR is an APPN end node or network node that owns dependent LUs but obtains SSCP services from a VTAM DLUS node.

Because a dependent LU server has both SSCP and APPN network node capabilities, it can provide both subarea and APPN session initiation flows for the served dependent LUs. The type of session flows used, subarea or APPN, depends on the location of the primary logical unit (PLU) in the network.

Use of the dependent LU server function requires, at a minimum:

- A dependent LU requester. IBM offers the following dependent LU requesters:
  - A VTAM-provided OS/2* dependent LU requester, which requires, at a minimum:
    - OS/2 V2
    - Communications Manager/2 V1.1 ServicePak WR06150, or the country equivalent for national language versions
  - IBM 3174 Establishment Controller Configuration Support C R5, or later

and, depending on your configuration, one or more of the following:

- IBM 3172 Interconnect Controller
- NCP V6R2

This function adds the following to this book:

- The following new operands on the PATH definition statement in the switched major node:
  - DLCADDR
  - DLURNAME.

- A new MAXDLUR operand on the VBUILD definition statement in the switched major node
- A new CPSVRMGR logon mode on the LOGMODE operand in the IBM-supplied default logon mode table.

Dynamic Reconfiguration Enhancement

In pre-V4R2 releases, you can use only the VARY DRDS command, the MODIFY DR command, and implicit dynamic reconfiguration to dynamically reconfigure your network. Because these methods do not modify the VTAMLST definition file, the changes can be lost when VTAM is restarted or when NCP is reactivated.

Now, you can dynamically reconfigure your network by making changes in the VTAMLST definition file and issuing a VARY ACT command with the new UPDATE operand. VTAM updates the current configuration to match the new configuration in the VTAMLST definition file. The major node can stay active; only the subordinate resources that you are changing must be inactive. Because the VTAMLST definition file is modified, the updated configuration is restored when VTAM or NCP is restarted.

This enhancement to dynamic reconfiguration allows VTAM and NCP to remain synchronized even when a resource is deleted from and added back to the same location or moved from and moved back to the same location.

In addition, VTAM now enables you to change certain operand values on PU and LU definition statements in NCP, local SNA, and switched major nodes. You do this by making the changes in the VTAMLST definition file and using the new UPDATE operand on the VARY ACT command for the major node. You do not need to deactivate the entire major node to change the operand values; only the affected resources must be inactive.

Use of the new dynamic reconfiguration function requires, at a minimum, NCP V6R2 for:

- Moving resources from and back to the same location
- Deleting resources from and adding them back to the same location.
- Adding values to the GP3174 and RETRIES operands on the PU definition statement in the NCP major node.

See the relevant sections of this book for information about which operand values you can dynamically change on the PU and LU definition statements in the channel-attachment (SDLC nonswitched), local SNA, NCP, packet, and switched major nodes.

Expanded Addressing Pool

In pre-V4R2 releases, VTAM was limited to 65,535 host network element addresses. VTAM now increases this limitation to 16,776,960 (16384K) element addresses assigned to LU-LU sessions or DLUR-supported SSCP-PU or SSCP-LU sessions.

Note that this extension to the number of addresses applies to host subarea element addresses. NCP is still limited to 32K element addresses from its subarea.

This function requires, at a minimum, NPM V1R6 with appropriate program temporary fix (PTF) or NPM V2R1 with appropriate PTF, for NPM to collect data about resources with high network addresses.

Note: NetView* support is not available for the expanded addressing pool function.

See Chapter 4, "Start Options" for information about a new ENHADDR start option.

Network-Qualified Names

In pre-V4R2 releases, you need to ensure that all resources that are capable of LU-LU sessions within SNA interconnected networks have unique names. To avoid duplicate names for LUs of types other than type 6.2, you can use the following methods:

- Session management exit routine
- USERVAR
- NetView alias name translation facility
- Predefined cross-domain resources and adjacent SSCP tables to avoid routing.

Now, you can use network-qualified names to ensure that resources that are LU-LU session-capable within SNA interconnected networks 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. You can now use network-qualified names in operator commands and session requests. You can also now specify that a network identifier and a resource name function as a network-qualified name in programming macroinstructions.

VM Use of network-qualified names requires, at a minimum, VM/ESA V1R3.

This function adds the following to this book:

- A new NQNMODE operand on the GROUP definition statement
- New LUALIAS and NQNMODE operands on the CDRSC definition statement in the CDRSC major node
- Changes to the LOGAPPL operand, which appears on many definition statements in many major nodes
- Changes to the name field and ACBNAME operand on the APPL definition statement in the application program major node
- A new MSGLVL operand on the USSMSG macroinstruction in the USS table that defines VTAM operator messages
- A new SCAN value on the BUFFER operand on the USSMSG macroinstruction in the USS table that defines terminal operator messages
- Changes to the BUFFER and TEXT operands on the USSMSG macroinstruction in the USS table that defines terminal operator messages
- Changes to the APPLID operand on the LOGCHAR macroinstruction in the interpret table
- The following new start options:
  - MSGLEVEL

  - NQNMODE
- A new IDTYPE operand on the TRACE and NOTRACE start options
- Changes to the ID operand on the TRACE and NOTRACE start options
- Changes to the DATEFORM start option and TRACE and NOTRACE start options for buffer and I/O traces
- Samples of the following updated IBM-supplied default tables:
  - ISTINCDT
  - ISTINCNO
  - ISTMGC10.

PU Dial at Major Node Activation

VTAM now enables you to specify whether to dial the PUs that are activated as a result of a switched major node activation. Doing so prevents you from having to issue a VARY DIAL command for each PU individually after those PUs are activated.

See the Switched Major Node section for information about a new DWACT operand on the PU definition statement in the switched major node.

Sift-Down Effect in Application Program and Cross-Domain Resource Manager Major Nodes

A GROUP definition statement has been added to the application program major node and cross-domain resource major node. As a result, some operands can now be sifted to subordinate definition statements in those major nodes.

See the relevant sections of this book for information about the new GROUP definition statement in the application program major node and cross-domain resource major node, including which operands can now be sifted from the GROUP definition statement to subordinate definition statements in those major nodes.

Specifying Search Order for Locating Switched PUs

A new start option enables you to specify the order in which VTAM searches control point names and station identifiers to locate switched PUs.

See Chapter 4, “Start Options” for information about a new SWNORDER start option.

Virtual-Route-Based Transmission Groups

A virtual-route-based transmission group (VR-based TG) is a logical mapping of an APPN transmission group (TG) over all of the subarea virtual routes between two VTAM V4R2 domains within the same APPN subnetwork. This enables APPN route calculations to include subarea links.

The VR-based TG is reported to APPN topology like any other APPN TG, and can be assigned TG characteristics that allow topology and routing services to include the VR-based TG in APPN session routes. The VR-based TG can also be used to establish CP-CP sessions between two VTAM control points, which in turn allows topology exchange and directory services functions to be performed between the two VTAMs.

Although the VR-based TG is reported to topology, the underlying explicit routes (ERs) and virtual routes (VRs) remain as they are and are used to transport the actual LU-LU session traffic between the VTAM domains. Existing subarea VR selection logic is used to determine the appropriate route through the subarea network for any given LU-LU session.

In addition to the path information, an SSCP-SSCP session is required between the two VTAMs for use in obtaining subarea routing information. The VR-based TG will not be reported as being active until the SSCP-SSCP session is established.

This function adds the following to this book:

- The following new operands on the CDRM definition statement in the cross-domain resource manager major node:
  - CAPACITY
  - COSTBYTE
  - COSTTIME
  - NN
  - PDELAY
  - SECURITY VM
  - TGP
  - UPARM1
  - UPARM2
  - UPARM3
  - VRTG
  - VRTGCPCP
- The following new start options:
  - VRTG
  - VRTGCPCP.

Where to Find Information about VTAM

"Bibliography" on page 827 describes the books in the VTAM V4R2 library, arranged according to related tasks. The bibliography also lists the titles and order numbers of books related to this book or cited by name in this book.

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
- LAN major nodes **VM,VSE**
- Local non-SNA major nodes
- Local SNA major nodes
- LU group major nodes
- Model major nodes
- NCP major nodes
- Packet major nodes **VM,VSE**
- 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.

MVS,VM 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 Chapter 3, “Routing and Dynamic Reconfiguration” on page 463.

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

Refer to “Dynamic Reconfiguration and Change of Operands” in the *VTAM 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.
- Support for integrated communication adapter (ICA) VM,VSE
  - SDLC nonswitched lines
  - SDLC switched lines
  - BSC nonswitched lines.

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 Interconnect Controller. The minor nodes are the line groups, lines, and physical units attached to the port.

LAN major node VM,VSE ***:*** A local area network (LAN) major node defines a connection (port) between VTAM and a LAN. 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.

Packet major node VM,VSE ***:*** A packet major node defines a channel unit address pair that is configured as an X.25 port on a communication adapter. The minor nodes are the line groups and lines connected to the port, and, for SNA only, the physical units connected over the lines, and the logical units under the physical units.

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.

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 553), 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 485.

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 466.

- 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 474.

  – 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 470.

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 478.

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 495 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 504 for more details.

Tables and Session Awareness Data Filter

Brief descriptions of the tables and filter follow.

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 641 for information on the macroinstructions that define a subarea COS table and “APPN Class-of-Service Definitions” on page 644 for information on the statements that define an APPN

COS. Refer to “APPN and Subarea Class of Service Resolution” in the *VTAM Network Implementation Guide* for a discussion of how classes of service are used.

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 714 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 659 for information on coding the macroinstructions that define an interpret table. Refer to “Establishing and Controlling SNA Sessions” in the *VTAM 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 666 for information on the macroinstructions that define a logon mode table. Refer to “Establishing and Controlling SNA Sessions” in the *VTAM Network Implementation Guide* for information on how the table 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 690 for information on the macroinstructions that define a USS table. Refer to “Establishing and Controlling SNA Sessions” in the *VTAM Network Implementation Guide* for information on how the table is used.

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 683 for information on the macroinstructions that define a model name table. Refer to “Establishing and Controlling SNA Sessions” in the *VTAM Network Implementation Guide* for information on how the table is used.

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 638 for information on the macroinstructions that define an associated LU table. Refer to "Establishing and Controlling SNA Sessions" in the *VTAM Network Implementation Guide* for information on how the table is used.

X.21 CPS Retry Table VSE ***:*** The X.21 call progress signal (CPS) retry table allows X.21 network system users to specify the information needed to retry **non-retriable** call progress signals. For each signal, you can specify a retry delay and retry limit.

You can load the CPS number, its corresponding wait interval and the respective number of retries in the CPS retry table by using the CPSTAB start option or the MODIFY TABLE command. Non-retriable errors are tested for user-specified retries using the list created by the MODIFY TABLE command. The CPS found in the list will cause the dial sequence to be re-executed after the time specified for this CPS has elapsed. While the redial task waiting, the port is put back to be used for other calls.

Retriable signals can also be coded in the CPS table if the RETRYTO operand on the LINE statement is zero (0). A 0 value means that the adapter has **not** delayed before returning the signal to VTAM. This control of retriable signals allows VTAM to reuse the line during delay. It also allows you to code different retry limits and retry delays for each signal.

It is possible that a series of call attempts to the same station could fail with different call progress signals. Therefore, a total number of retries for all signals must be enforced. This limit is the minimum number of retries specified for any signal received on the given call establishment attempt.

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 678 for information on coding the macroinstructions that define a message-flooding prevention table.

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 (MVS)" on page 686 for information on the macroinstructions that define a SAW data filter. Refer to "Reducing Session

Awareness (SAW) Data" in the *VTAM Network Implementation Guide* for information on how the filter 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 *VTAM 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 "VARY ACT Command" in *VTAM 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) or the national characters @, #, or $
Second to eighth character	Alphabetical (A–Z), numerical (0–9), or the national characters @, #, or $.

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

If coded, the *name* field must begin in column 1 of the definition statement and must be followed by one or more blanks. 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*.

Format of Macroinstructions

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

Name	Operation	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 $.

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

The *operation* 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.

Rules for Coding Continued Lines

Code VTAM and NCP definition statements and VTAM macroinstructions in columns 1–71 of a line. 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. Enter a nonblank continuation character in column 72. The continuation character is not considered part of the definition statement.
3. Continue 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 the blank happens to coincide with that column.

   A comment line cannot follow a continuation line.
4. If you must continue on another line, proceed with rule 1 on page 11.
5. Definition statements and macroinstructions can be coded on as many lines as needed.
6. Comments can appear on every line of a continued definition statement or macroinstruction.
7. 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:

TRACE **VSE**
VTAM
VTAMSEG
VTAMTERM.

The names of major node members need to follow the rules for names detailed in “Format of Definition Statements” on page 10 and “Format of Macroinstructions” on page 11.

Defining a Logical Unit under VM

When you define a logical unit that logs on to VM, its name cannot be the same as a VM user ID in the system.

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 10, 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.

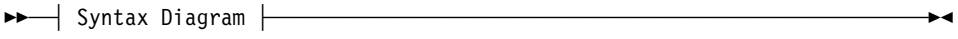
You can use comments in groups of definition statements for all types of major nodes and PATH definition statements. However, do not use macroinstructions or operands, such as GENEND, especially in a continued line of comments.

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.

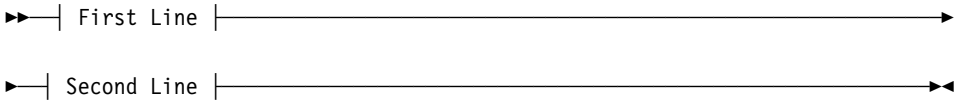
How to Read Railroad Track Syntax Diagrams

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 (►◄).



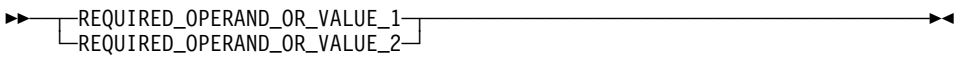
- If a diagram is longer than one line, the first line ends with a single arrowhead (►) and the second line begins with a single arrowhead.



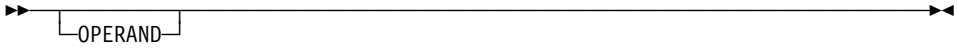
- Required operands and values appear on the main path line.

  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.

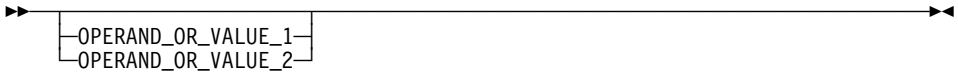


- Optional operands and values appear below the main path line.

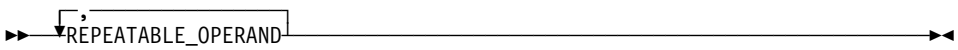


  You can choose not to code optional operands and values.

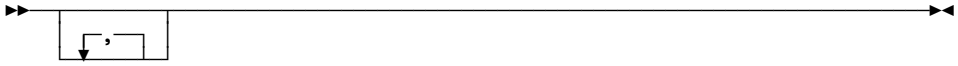
  If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.



- 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.



- 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.

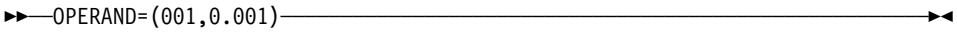


- A word in all uppercase is an operand or value you must spell exactly as shown. In this example, you must code ***OPERAND***.

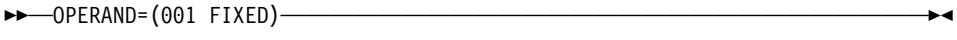
  Note: VTAM commands are not case sensitive. You can code them in uppercase or lowercase.

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

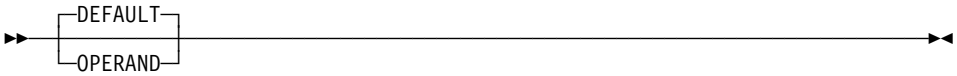
- If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code ***OPERAND=(001,0.001)***.



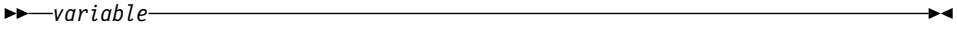
- If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code ***OPERAND=(001 FIXED)***.



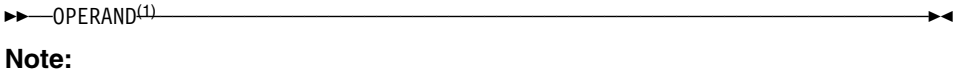
- Default operands and values appear above the main path line. VTAM uses the default if you omit the operand entirely.



- 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.

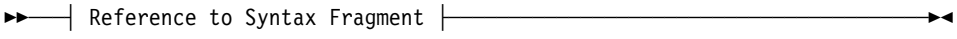


- References to syntax notes appear as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

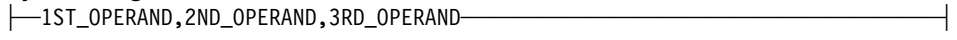


Note:
1 An example of a syntax note.

- 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.



Syntax Fragment:



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 478 for more information.

3. **MVS** 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.

   VM Store network definitions as sequential CMS files with a file type of VTAMLST. The files must have characteristics of `RECFM=F,LRECL=80`. The files are stored on a minidisk that is linked to the VTAM and VSCS virtual machine.

   Note: Whenever you change VTAM definitions under the VM operating system, you must use the GCS ACCESS command to reaccess the disk that contains the corresponding VTAMLST files before you activate the resources or restart VTAM.

   VSE Store network definitions in a private definition library referenced in the VTAM start procedure. 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 CPS retry (VSE), 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:

- MVS A library associated with the VTAMLIB DD statement.
- VM Refer to “Installing Tables and Modules in VM” in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit the table.
- VSE A private definition library referenced in the VTAM start procedure.

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:
   - MVS As a member in SYS1.VTAMLST. The member name is used as the name of the table.
   - VM With a filetype of VTAMLST.
   - VSE In a private definition library referenced in the VTAM start procedure.

Chapter 2. 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
- LAN VM,VSE
- Local non-SNA
- Local SNA
- LU group
- Model
- Network Control Program (NCP)
- Packet VM,VSE
- 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	26	R			APPN only
name	ADJCP	DYNLU	23			DYNLU start option	APPN only
		NATIVE	24				BN=YES; APPN only
		NETID	25				APPN only
		NN	25				APPN only VN
		VN	26			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.

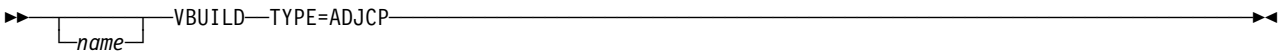
NODETYPE Start Option: Note that the NODETYPE start option is required for this major node.

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.

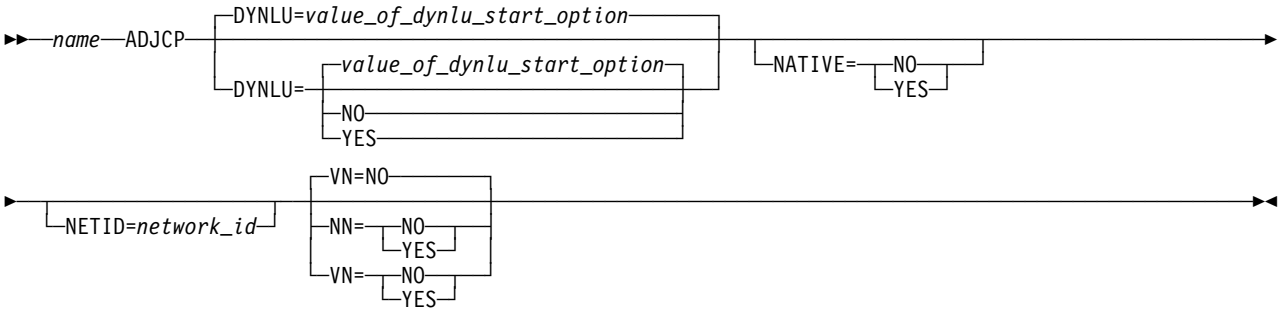
Full Syntax

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

VBUILD



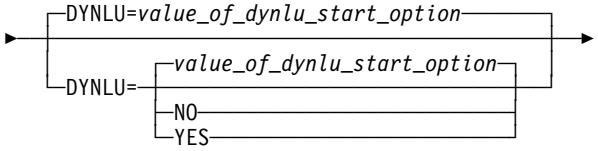
ADJCP



Operand Descriptions

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

DYNLU



statement: ADJCP
dependency: 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

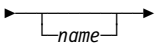
adjacent link station (ALS) is assigned to this adjacent CP and to all resources attached to the ALS.

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



statements: VBUILD

statements: ADJCP

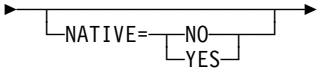
On VBUILD: specifies the optional name of this major node.

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



statements: ADJCP
dependency: 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.

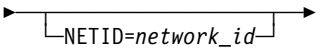
- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

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



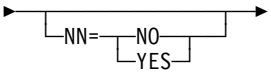
statement: ADJCP
dependency: APPN only

is 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



statement: ADJCP
dependency: VN; APPN only

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

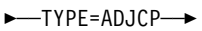
If you specify a value for NN, it is validated when a connection is attempted to the adjacent node. If the adjacent CP is not the type of node that is expected, then connection setup fails. If you do not specify a value for NN, then the APPN capabilities of the adjacent node are identified and accepted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

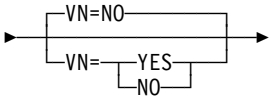
TYPE



statement: VBUILD
dependency: APPN only

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

VN



statements: ADJCP
dependency: NN; DYNLU=NO; APPN only

specifies that a virtual node is to be predefined.

Note: NN and VN are mutually exclusive.

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.

MVS 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 *VTAM Network Implementation Guide* for information on how TSO/VTAM applications are registered.

VM VSCS is a VTAM application program and must be defined to VTAM as an application program major node. For specific information on defining VSCS to VTAM, refer to “Defining VSCS as a VTAM Application Program” in the *VTAM Network Implementation Guide*.

Non-LU 6.2 Application Programs

Table 3 (Page 1 of 2). 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	53	R			
name	GROUP	See the APPL definition statement for operands.					
name	APPL	ACBNAME	32			*name* value	
		APPC	33		G	NO	
		ASLENT	33		G		ASLTAB
		ASLTAB	33		G		

Table 3 (Page 2 of 2). Application Program Major Node (Non-LU 6.2) Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		AUTH	34		G	(NOACQ, NOASDP, NOCNM, NOPASS, NOPO, NOTSO, VPACE)	MVS,VM
		AUTHEXIT	36		G	NO	VM
		CMPAPPLI	37		G	0	
		CMPAPPLO	37		G	0	
		DLOGMOD	38		G		MODETAB
		EAS	41		G	509	
		ENCR	41		G	OPT	MVS
		FASTPASS	42		G	YES	TSO only
		HAVAIL	43		G	NO	MVS
		LOSTERM	44		G	NORMAL	
		MAXPVT	44		G	0	VM
		MDLENT	45		G		MDLTAB
		MDLTAB	45		G		
		MODETAB	46		G	ISTINCLM	
		PARSESS	48		G	NO	APPC=NO
		POAQLIM	48		G		AUTH
		PRTCT	49		G		
		REGISTER	49		G	CDSERVR	APPN only
		SESSLIM	51		G	NO	
		SONSCIP	51		G	NO	
		SPAN	52		G		NetView
		SRBEXIT	52		G	NO	MVS
		SSCPFM	52		G	USSPOI	AUTH
		USSTAB	53		G	ISTINCNO	AUTH
		VCNS	54		G	NO	MVS,VM
		VPACING	55		G	7	
		VTAMFRR	56		G	NO	MVS

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 (Page 1 of 3). 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	53	R			
name	GROUP	See the APPL definition statement for operands.					
name	APPL	ACBNAME	32			*name* value	
		APPC	33		G	NO	APPC=YES
		ASLENT	33		G		ASLTAB
		ASLTAB	33		G		

Table 4 (Page 2 of 3). Application Program Major Node (LU 6.2) Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		ATNLOSS	34		G	LAST	APPC=YES
		AUTH	34		G	(NOACQ, NOASDP, NOCNM, NOPASS, NOPO, NOTSO, VPACE)	MVS,VM
		AUTHEXIT	36		G		VM
		AUTOSES	36		G	0	APPC=YES
		CMPAPPLI	37		G	0	
		CMPAPPLO	37		G	0	
		DDRAINL	38		G	NALLOW	APPC=YES
		DLOGMOD	38		G		MODETAB
		DMINWNL	39		G	1	APPC=YES DMINWNR DSESLIM
		DMINWNR	39		G	1	APPC=YES DMINWNL DSESLIM
		DRESPL	39		G	NALLOW	APPC=YES
		DSESLIM	40		G	2	APPC=YES; DMINWNL; DMINWNR
		DSPLYWLD	40		G	YES	AUTH
		EAS	41		G	509	
		ENCR	41		G	OPT	MVS
		FASTPASS	42		G	YES	TSO only
		HAVAIL	43		G	NO	MVS
		LIMQSINT	43		G		APPC=YES
		LMDENT	43		G	19	APPC=YES
		LOSTERM	44		G	NORMAL	
		MAXPVT	44		G	0	VM
		MDLENT	45		G		MDLTAB
		MDLTAB	45		G		
		MODETAB	46		G	ISTINCLM	
		OPERCNOS	47		G	NALLOW	APPC=YES
		PARSESS	48		G	YES	APPC=YES
		POAQLIM	48		G		AUTH
		PRTCT	49		G		
		REGISTER	49		G	CDSERVR	APPN only
		SECACPT	49		G	NONE	APPC=YES
		SECLVL	50		G	ADAPT	MVS,VM APPC=YES
		SESSLIM	51		G	NO	
		SONSCIP	51		G	NO	
		SPAN	52		G		NetView
		SRBEXIT	52		G	NO	MVS
		SSCPFM	52		G	USSPOI	AUTH
		SYNCLVL	53		G	CONFIRM	APPC=YES
		USSTAB	53		G	ISTINCNO	AUTH
		VCNS	54		G	NO	MVS,VM
		VERIFY	54		G	NONE	MVS,VM APPC=YES
		VPACING	55		G	7	

Table 4 (Page 3 of 3). Application Program Major Node (LU 6.2) Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		VTAMFRR	56		G	NO	MVS

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.

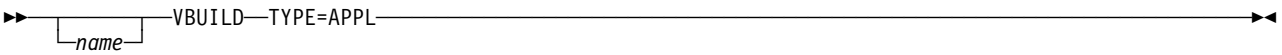
APPL Definition Statement: Code one APPL definition statement for each application program that is in the VTAM domain and include it in an application program major node.

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 “Defining Application Programs” in the *VTAM Network Implementation Guide* for specific information on these requirements.

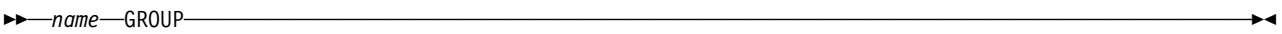
Full Syntax

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

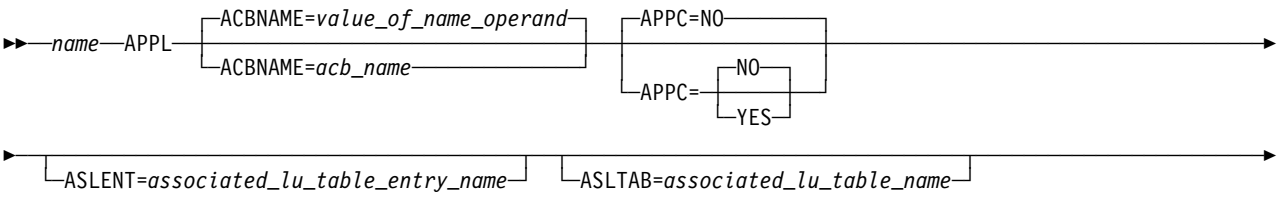
VBUILD

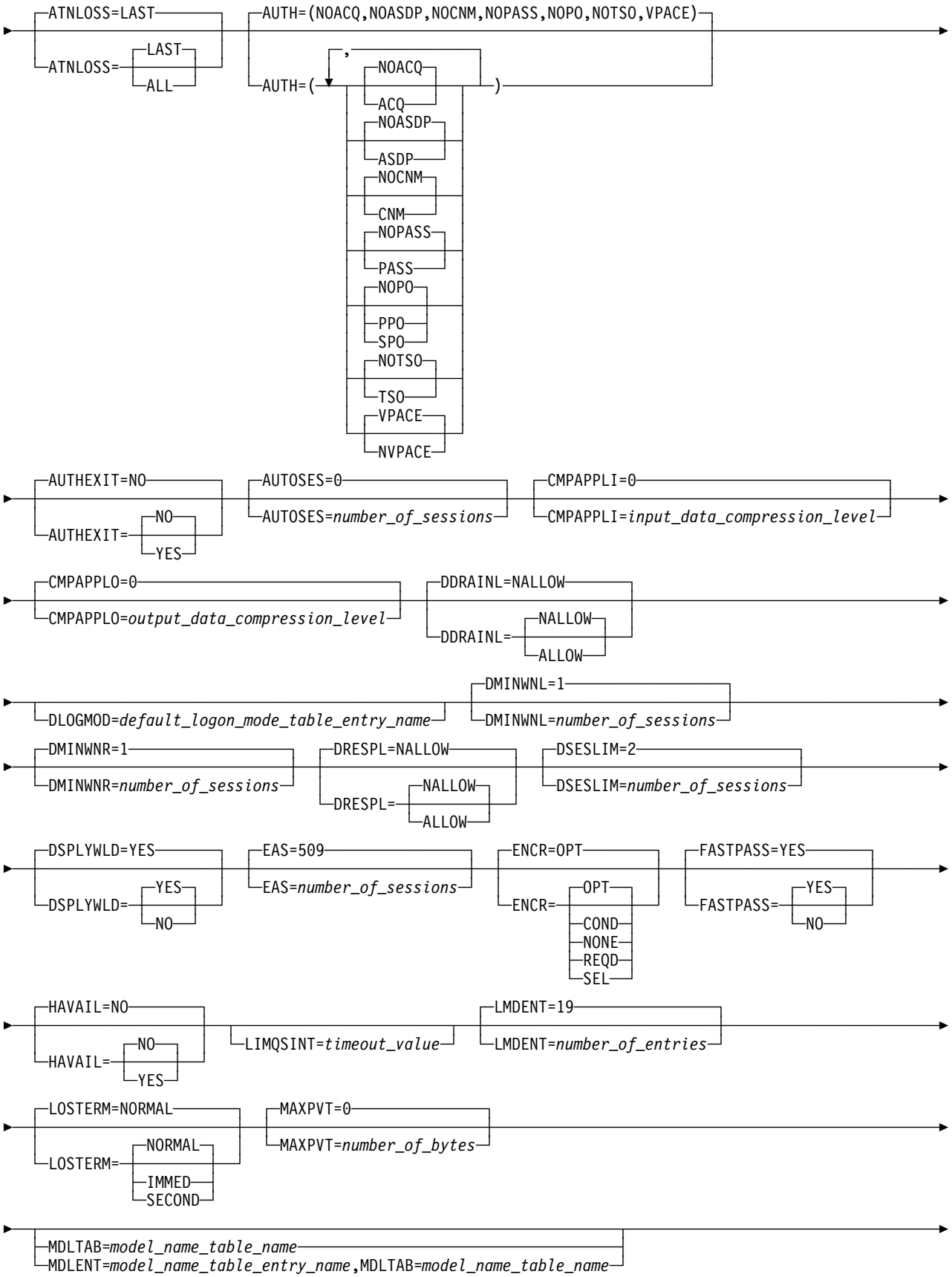


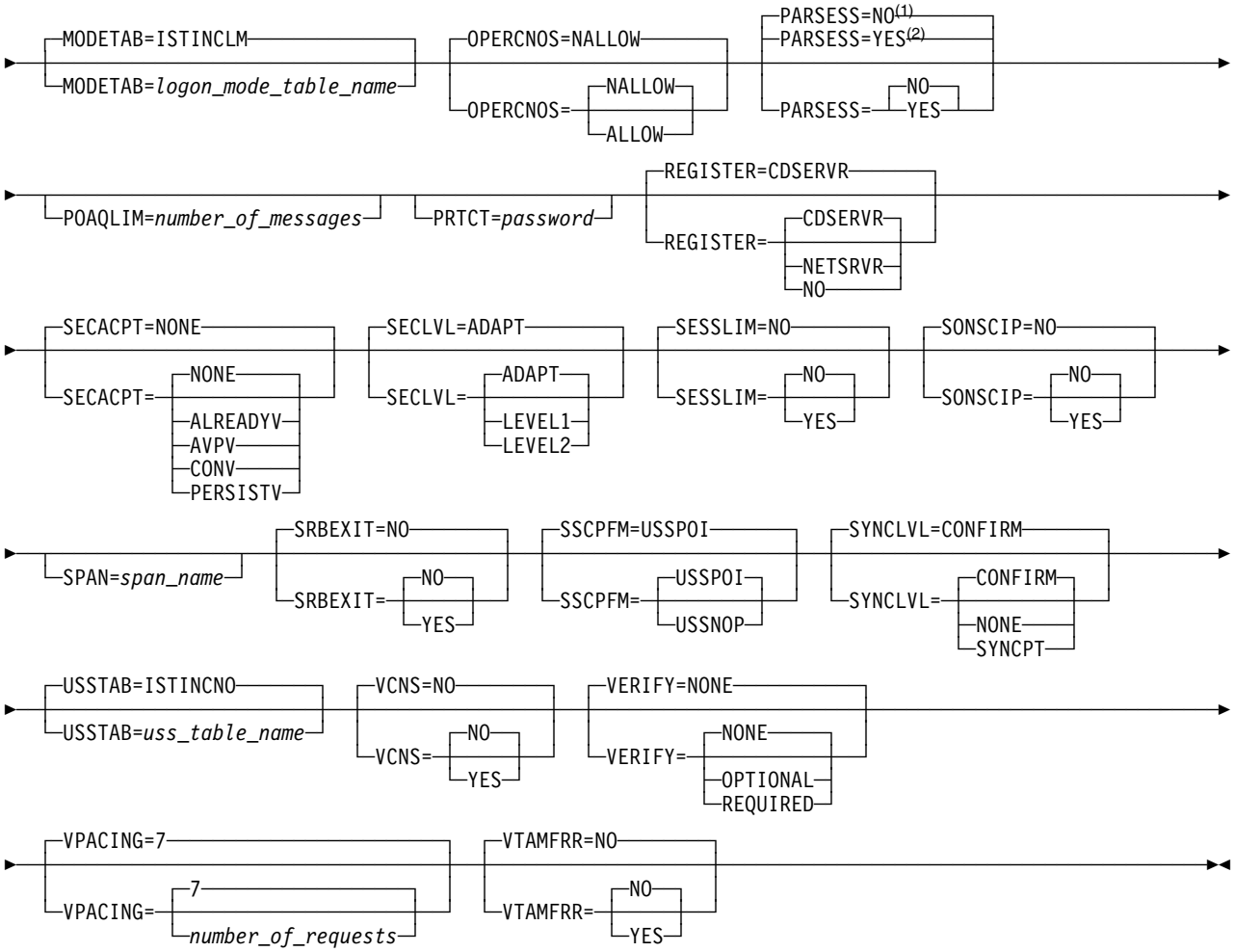
GROUP



APPL




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
)
AUTHEXIT=NO
AUTHEXIT=
NO
YES
AUTOSES=0
AUTOSES=number_of_sessions
CMPAPPLI=0
CMPAPPLI=input_data_compression_level
CMPAPPLO=0
CMPAPPLO=output_data_compression_level
DDRAINL=NALLOW
DDRAINL=
NALLOW
ALLOW
DLOGMOD=default_logon_mode_table_entry_name
DMINWNL=1
DMINWNL=number_of_sessions
DMINWNR=1
DMINWNR=number_of_sessions
DRESPL=NALLOW
DRESPL=
NALLOW
ALLOW
DSESLIM=2
DSESLIM=number_of_sessions
DSPLYWLD=YES
DSPLYWLD=
YES
NO
EAS=509
EAS=number_of_sessions
ENCR=OPT
ENCR=
OPT
COND
NONE
REQD
SEL
FASTPASS=YES
FASTPASS=
YES
NO
HAVAIL=NO
HAVAIL=
NO
YES
LIMQSINT=timeout_value
LMDENT=19
LMDENT=number_of_entries
LOSTERM=NORMAL
LOSTERM=
NORMAL
IMMED
SECOND
MAXPVT=0
MAXPVT=number_of_bytes
MDLTAB=model_name_table_name
MDLENT=model_name_table_entry_name,MDLTAB=model_name_table_name



Notes:

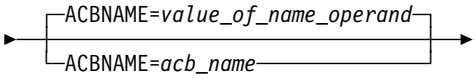
1 PARSESS defaults to NO when APPC=NO.

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



statements: APPL
dependency: name

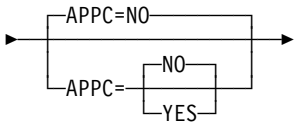
specifies the minor node name assigned to this application program. This name must be unique within the domain in which the application program resides.

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.

APPC



statements: GROUP, APPL
dependency: none

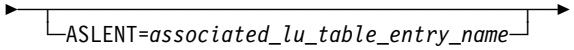
specifies whether the application program can issue APPCCMD macroinstructions. Refer to Chapter 14, “LU 6.2 Macroinstruction Syntax and Operands” in *VTAM Programming for LU 6.2* for a description of this macroinstruction.

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

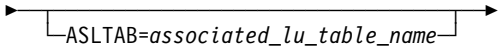


statements: GROUP, APPL
dependency: ASLTAB

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



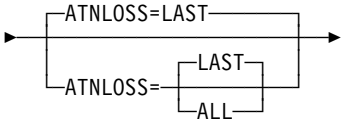
statements: GROUP, APPL
dependency: none

specifies the name of the associated LU table to be used for this application program. See “Associated LU Table” on page 638 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about this command.

ATNLOSS



statements: GROUP, APPL
dependency: APPC=YES

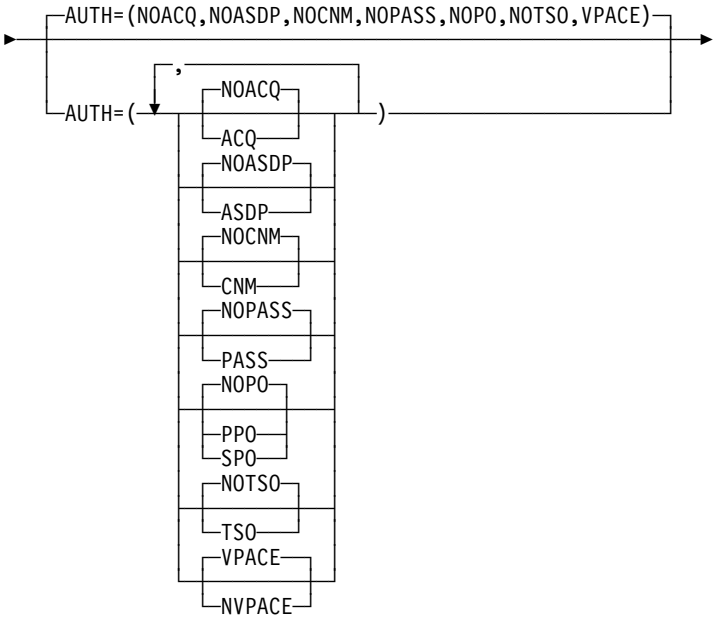
specifies when the ATTN exit should be scheduled.

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 (MVS, VM)



statements: GROUP, APPL
dependency: 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 Chapter 12, "Coding for the Communication Network Management Interface" in *VTAM 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 "CNM Routing Table (MVS)" in *VTAM 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 "CLSDST OPTCD=PASS" in *VTAM Programming*.

AUTH=PPO
AUTH=SPO
AUTH=NOPO
specifies whether the application program can issue VTAM operator commands and receive responses and unsolicited messages. Appendix K, "Program Operator Coding Requirements" in *VTAM 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.

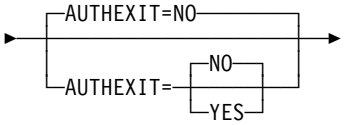
AUTH=TSO
AUTH=NOTSO
MVS specifies whether the application program is a TSO/VTAM time-sharing program. If it is a TSO/VTAM time-sharing program, code TSO. If not, code NOTSO or take the default.

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: MVS 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 “Session Pacing” in the *VTAM Network Implementation Guide*.

AUTHEXIT (VM)



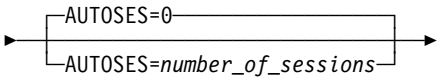
statements: GROUP, APPL
dependency: none

specifies whether the application program’s exit routines are to run in supervisor state. If AUTHEXIT=YES, they will run in supervisor state if the application is authorized by group control system (GCS) to do so. Before returning control to GCS, the application must clean up any recovery environment (ESTAE) it sets up.

AUTHEXIT=NO
specifies that the application program’s exit routines do not run in supervisor state.

AUTHEXIT=YES
specifies that the application program’s exit routines do run in supervisor state.

AUTOSES

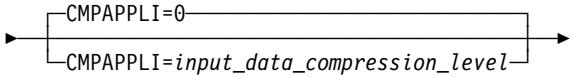


statements: GROUP, APPL
dependency: APPC=YES
range: 0–32767

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 “Session Activation” in *VTAM Programming for LU 6.2* for more information about the role of the AUTOSES operand in session activation.

CMPAPPLI



statements: GROUP, APPL
dependency: 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

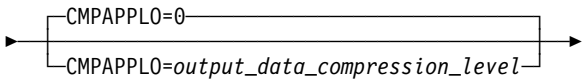
The limit specified by the operand is enforced only during sessions for which:

- The application acts as the PLU.
- The limit is lower than that specified by the CMPVTAM start option.

Note: VM The limit of the CMPVTAM start option overrides the values of 2, 3, and 4. You can specify 2, 3, or 4 as values for CMPAPPLI, but these values have the same effect as 1. These values are for operating system architecture compatibility.

Refer to “Data Compression” in the *VTAM Network Implementation Guide* for more information about compression levels.

CMPAPPLO



statements: GROUP, APPL
dependency: 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:

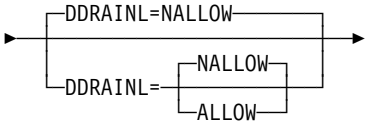
0 No compression
1 Run-length encoding (RLE)
2 Small adaptive compression table
3 Medium adaptive compression table
4 Large adaptive compression table

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.

Note: VM The limit of the CMPVTAM start option overrides the values of 2, 3, and 4. You can specify 2, 3, or 4 as values for CMPAPPLO, but these values have the same effect as 1. These values are for operating system architecture compatibility.

Refer to the *VTAM Network Implementation Guide* for more information about compression levels.

DDRAINL



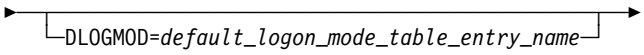
statements: GROUP, APPL
dependency: APPC=YES

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



statements: GROUP, APPL
dependency: MODETAB

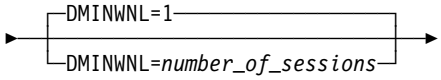
specifies the name of the logon mode table entry used by default if one is not otherwise provided. If you do not code this operand and the name of a logon mode table entry is not otherwise provided, VTAM uses the first entry in the applicable logon mode table (specified on the MODETAB operand or used 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 666.

DMINWNL



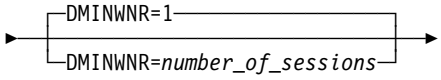
statements: GROUP, APPL
dependency: APPC=YES, DMINWNR, DSESLIM
range: 0–32767

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 "LU 6.2 Session Limits" in the *VTAM Network Implementation Guide* and "APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN" in *VTAM Programming for LU 6.2* for information on how VTAM negotiates the actual number of contention-winner sessions for this application (local logical unit).

DMINWNR



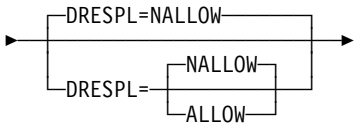
statements: GROUP, APPL
dependency: APPC=YES, DMINWNL, DSESLIM
range: 0–32767

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 "LU 6.2 Session Limits" in the *VTAM Network Implementation Guide* and "APPCCMD CONTROL=ALLOC, QUALIFY=CONWIN" in *VTAM Programming for LU 6.2* for information on how VTAM negotiates the actual number of contention-winner sessions for the remote logical unit.

DRESPL



statements: GROUP, APPL
dependency: APPC=YES

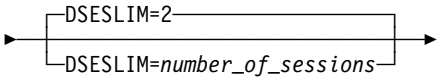
specifies whether the local application program (local logical unit) will deactivate sessions on receipt of a CNOS request that specifies the application program as the responsible logical unit.

Refer to “Draining and Session Deactivation Responsibility” in *VTAM Programming for LU 6.2* 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



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

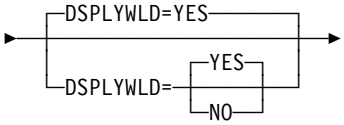
defines the maximum number of sessions to be allowed between the local application logical unit and a remote logical unit on a given mode name group.

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 “Managing LU 6.2 Sessions with Operator Commands” in the *VTAM Network Implementation Guide* and “How Defined Negotiation Limits Are Set” in *VTAM Programming for LU 6.2* for information on how VTAM determines the negotiated session limit.

DSPLYWLD



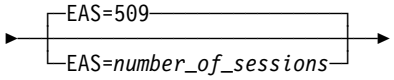
statements: GROUP, APPL
dependency: AUTH=PPO or AUTH=SPO

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



statements: GROUP, APPL
dependency: 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.

MVS 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.

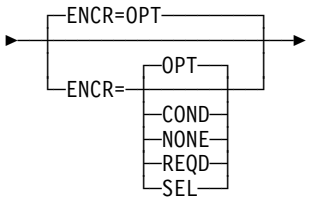
Note: You should code EAS on applications that have only a few concurrent sessions, to conserve storage.

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 (MVS)



statements: GROUP, APPL
dependency: none

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

ENCR=COND
specifies that this application program supports mandatory encryption (see ENCR=REQD) if both sides support encryption. If the secondary logical unit does not support encryption, the session does not fail; instead, a session without encryption is established.

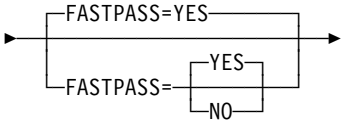
ENCR=NONE
specifies that this application program cannot have cryptographic sessions.

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.

FASTPASS (MVS)



statements: GROUP, APPL
dependency: 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.

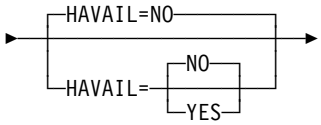
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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 (MVS)



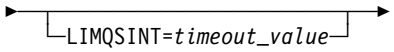
statements: GROUP, APPL
dependency: none

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



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

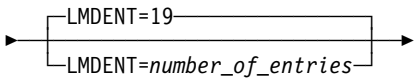
specifies how often VTAM searches the queue for unused LU 6.2 contention winners.

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

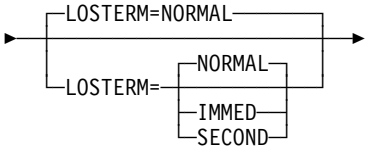


statements: GROUP, APPL
dependency: APPC=YES
range: 0–65535

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



statements: GROUP, APPL
dependency: none

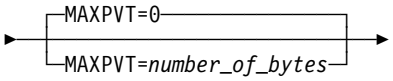
specifies how VTAM terminates a session with a hung LU. You can force an LU to an inactive state when it becomes hung during a session with an application program that has a LOSTERM exit routine but no NSEXIT exit routine.

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.

MAXPVT



statements: GROUP, APPL
dependency: 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).

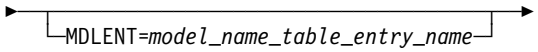
MAXPVT=0
specifies that no limit is defined and thus, when no storage is available in the address space associated with this application program, the session is cleared.

VM When this occurs, the application program's LOSTERM exit routine is scheduled with a reason code of 36 (X'24').

MAXPVT=*number_of_bytes*
MAXPVT=*number_of_bytes***K**
number_of_bytes is 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. VM If storage is not available, the session is cleared and the application program's LOSTERM exit routine is scheduled with a reason code of 36 (X'24').

MAXPVT=*number_of_bytes***M**
number_of_bytes is 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.

MDLENT

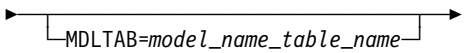


statements: GROUP, APPL
dependency: MDLTAB

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



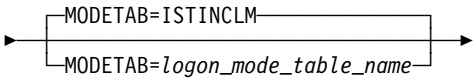
statements: GROUP, APPL
dependency: none

specifies the name of the model name table to be used for this application program. See "Model Name Table" on page 683 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 714 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 "MODIFY TABLE Command" in *VTAM Operation* for more information about the MODIFY TABLE command.

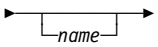
MODETAB



statements: GROUP, APPL
dependency: 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 666. 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.

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 “Naming an Application Program” in the *VTAM 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).

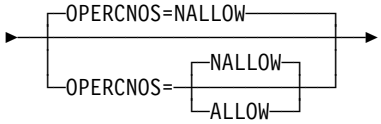
A logical unit using the application program's network name (the *name* operand on the APPL definition statement) can establish a session with the application program from any domain in the network. A logical unit using the ACB name to establish a session with an application program must be in the same 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. "ACB-Create an Access Method Control Block" in *VTAM Programming* describes the ACB macroinstruction.

MVS Refer to "Defining the TCAS Application to VTAM" and "Defining TSO/VTAM Session Parameters" in the *VTAM 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



statements: GROUP, APPL
dependency: APPC=YES

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

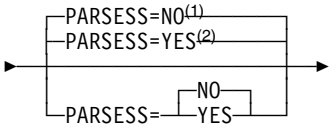
The MODIFY CNOS command changes the CNOS parameters governing LU 6.2 sessions. The MODIFY DEFINE command establishes new entries or modifies existing entries in the LU-mode table for the specified application program.

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



Notes:
1 PARSESS defaults to NO when APPC=NO.
2 PARSESS defaults to YES when APPC=YES.

statements: GROUP, APPL
dependency: APPC=YES

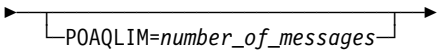
specifies whether this application program can have multiple concurrent (parallel) sessions with another application program or independent logical unit in the same domain or in another domain.

PARSESS=NO
restricts this application program to one LU-LU session, between itself and another logical unit, at a time. Note that if you code PARSESS=NO and APPC=YES, VTAM supplies an override value of YES for PARSESS.

PARSESS=YES
allows this application program to have multiple LU-LU sessions between the same session partners. Note that if you code PARSESS=YES and APPC=NO (or let it default), VTAM supplies an override value of NO for PARSESS.

Refer to “Parallel Sessions and Element Addressing” in the *VTAM Network Implementation Guide* for a discussion of parallel sessions in a multiple network environment.

POAQLIM



statements: GROUP, APPL
dependency: 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.

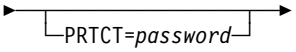
POAQLIM applies only to a program operator application program (AUTH=PPO or AUTH=SPO).

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

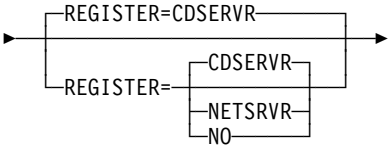


statements: GROUP, APPL
dependency: none

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.

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

REGISTER



statements: GROUP, APPL
dependency: APPN only

specifies how a resource should be registered.

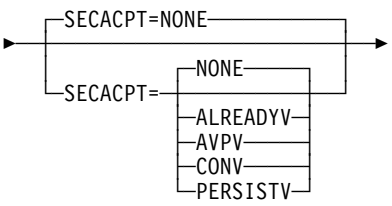
REGISTER=CDSERVR
indicates that an end node resource is registered to a network node server and directory resource registration is requested for it. A network node resource is registered at the central directory server. If the node is configured as a central directory server, this value has the same 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: **MVS** 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



statements: GROUP, APPL
dependency: APPC=YES

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

SECACPT=ALREADYV MVS,VM
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.

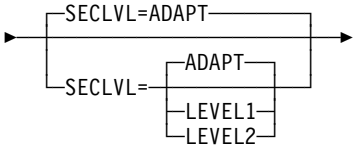
SECACPT=AVPV MVS
specifies that this logical unit supports conversation requests containing access security subfields. The logical unit also accepts the already-verified indications and the persistent verification indications that it receives in conversation requests.

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 MVS
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 (MVS, VM)



statements: GROUP, APPL
dependency: 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.

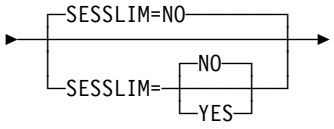
This operand is valid only when APPC=YES and session-level LU-LU verification is used. (VERIFY=OPTIONAL or VERIFY=REQUIRED must be coded on this application program major node. If you code VERIFY=NONE or let it default, a parameter conflict results.)

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



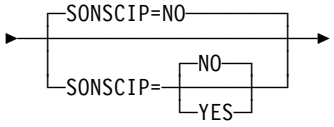
statements: GROUP, APPL
dependency: none

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



statements: GROUP, APPL
dependency: none

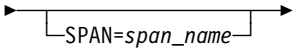
determines whether the application program receives UNBIND RUs in its SCIP exit routine when the application program is acting as the primary end of a failing session.

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 "Receiving an UNBIND Request" in *VTAM 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

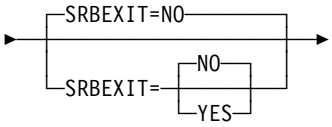


statements: GROUP, APPL
dependency: NetView

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 (MVS)



statements: GROUP, APPL
dependency: none

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

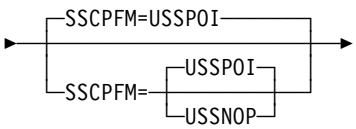
SRBEXIT=NO
specifies that the application program is not authorized to use SRB processing in its exit routines.

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.

Note: If multiple applications open an ACB under one task, VTAM recognizes the SRBEXIT operand for the first application that opens the ACB. Therefore, the first open ACB determines whether exit routines operate in SRB or task control block (TCB) mode.

This operand is ignored for subsequent applications opening ACBs under the same task. If you code SRBEXIT=YES on the first application opening the ACB, the subsequent applications opening ACBs under the same task must be authorized.

SSCPFM



statements: GROUP, APPL
dependency: AUTH=PPO or AUTH=SPO

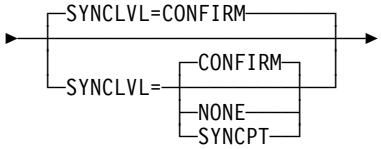
specifies the default USS table that is used to send a VTAM operator message to the program operator if the message is not included in a user-defined table. SSCPFM has no effect on the USS table that is used to process VTAM operator commands.

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



statements: GROUP, APPL
dependency: APPC=YES

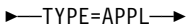
specifies the application program's synchronization support level.

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.

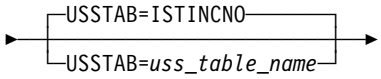
TYPE



statements: VBUILD
dependency: none

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

USSTAB



statements: GROUP, APPL
dependency: AUTH=PPO or AUTH=SPO

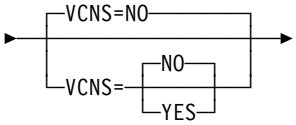
specifies a user-defined USS table that contains user-modified VTAM operator commands and VTAM operator messages to be used by the program operator.

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 (MVS, VM)



statements: GROUP, APPL
dependency: none

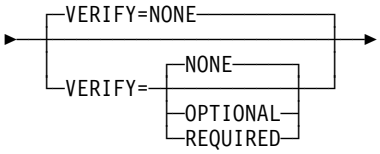
specifies whether VTAM is allowed to perform VTAM Common Network Services (VCNS) functions.

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 (MVS, VM)



statements: GROUP, APPL
dependency: 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 “Security Management Product Requirements (MVS, VM)” in the *VTAM Programming for LU 6.2* for more information.

VM When a security management product is installed and active, the virtual machine's RACROUTE interface must be initialized even if the application program is not using VTAM security services. Refer to Chapter 12, “VTAM's LU 6.2 Security Options” in *VTAM Programming for LU 6.2* for details on initializing the RACROUTE interface, and to “VTAM Application Security ” in the *VTAM Network Implementation Guide* for general information on RACF.

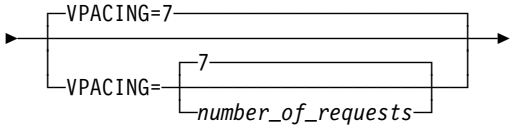
VERIFY=NONE
specifies that no verification of the partner LU’s identity takes place during session activation.

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 “LU 6.2 Security” in the *VTAM 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



statements: GROUP, APPL
dependency: 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.

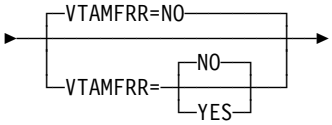
If the value is greater than 63, the default value is used. No pacing of requests to the application program is done if VPACING=0.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

VTAMFRR (MVS)



statements: GROUP, APPL
dependency: 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.
- Integrated communication adapter (ICA) VM,VSE

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 68 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 (Page 1 of 2). Channel-Attachment (CTCA) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	113	R			
		CONFGDS	79				MVS,VSE
		CONFGPW	79				MVS,VSE
							CONFGDS
name	GROUP	LNCTL=CTCA	90	R			
		REPLYTO	105			3.0	MIH
		SPAN	108				NetView
name	LINE	ADDRESS	74				
		ISTATUS	89		G	ACTIVE	
		MAXBFRU	93		G	1	
		MIH	97		G	NO	LNCTL=SDLC
		SPAN	108				NetView

Table 5 (Page 2 of 2). Channel-Attachment (CTCA) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PU	DELAY	82		G,L	0.100	
		ISTATUS	89		G	ACTIVE	
		PUTYPE	103		G,L	4	
		SPAN	108				NetView
		TGN	111			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 68 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 (Page 1 of 2). Channel-Attachment (MPC) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	113	R			
		CONFGDS	79				MVS,VSE
		CONFGPW	79				MVS,VSE CONFGDS
name	GROUP	LNCTL=MPC	90	R			
		REPLYTO	105			3.0	
		SPAN	108				NetView
name	LINE	READ	104	R			LNCTL=MPC

Table 6 (Page 2 of 2). Channel-Attachment (MPC) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		WRITE	115	R			LNCTL=MPC
		ISTATUS	89		G	ACTIVE	
		MAXBFRU	93		G	1	
		SPAN	108				NetView
name	PU	ISTATUS	89		G	ACTIVE	
		PUTYPE	103		G,L	4	
		SPAN	108				NetView
		TGN	111			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 68 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 channel unit address 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 “Sharing NCP Peripheral Nodes (CMC Configurations)” in the *VTAM 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	113	R			
		CONFGDS	79				MVS,VSE
		CONFGPW	79				MVS,VSE
							CONFGDS
name	GROUP	LNCTL=NCP	90	R			
		SPAN	108				NetView
name	LINE	ADDRESS	74				
		ISTATUS	89		G	ACTIVE	
		MAXBFRU	93		G	3	
		SPAN	108				NetView
name	PU	CHANCON	78		G,L	COND	LNCTL=NCP
		DELAY	82		G,L	0.200	
		ISTATUS	89		G	ACTIVE	
		MAXDATA	95		G,L	65535	LNCTL=NCP
		PUTYPE	103		G,L	4	
		SPAN	108				NetView
		TGN	111			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.

Integrated Communication Adapter Line Support (VM, VSE)

You can define one of three types of integrated communication adapter (ICA) support in a channel-attachment major node:

- SDLC nonswitched
- SDLC switched
- BSC nonswitched

SDLC Nonswitched Lines

To define ICA support for SDLC nonswitched lines, code the following definition statements:

```
VBUILD
GROUP LNCTL=SDLC, DIAL=NO
LINE
PU
```

LU

VBUILD Definition Statement: See "VBUILD Definition Statement" on page 68 for details on the VBUILD definition statement.

GROUP Definition Statement: Code a GROUP definition statement to define an SDLC nonswitched line group.

Note that 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 SDLC nonswitched line.

PU Definition Statement: Code one PU definition statement for each physical unit of any type (1, 2, 4, or 5) with which VTAM communicates over this SDLC nonswitched link.

VTAM uses the PU definition statement to define the link station for an adjacent host processor (physical unit type 5) or communication controller (physical unit type 4).

Code only one PU definition statement for each LINE definition statement if you code PUTYPE=4 or PUTYPE=5 on the PU definition statement.

LU Definition Statement: Code one LU definition statement for each logical unit associated with an SDLC station (type 1 or 2 physical unit) attached to an SDLC nonswitched line.

Table 8 (Page 1 of 3). Channel-Attachment (SDLC Nonswitched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	113	R			
		CONFGDS	79				VSE
		CONFGPW	79				VSE CONFGDS
name	GROUP	DIAL=NO	82	R			nonswitched lines only
		LNCTL=SDLC	90	R			
		SPAN	108				NetView
name	LINE	ACTIVTO	74		G	60	REPLYTO
		ADDRESS	74			030	
		BUFCAP	77		G	1	
		CORNUM	80				
		ISTATUS	89		G	ACTIVE	
		LPDATS	92		G	NO	
		MAXBFRU	93		G	(2,8)	
		MODE	98		G	PRI	
		PAUSE	102		G	0.1	not valid for PU type 4
		PORT	103				
		REPLYTO	105		G	1.0	
		SERVLIM	107		G	4	
		SPAN	108				NetView
name	PU	ADDR	74	R			PU types 1, 2

Table 8 (Page 2 of 3). Channel-Attachment (SDLC Nonswitched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		AUTHLEN	76			YES	APPN only PU Type 2.1
		CAPACITY	78		G,L	8K	APPN only
		CONNTYPE	79		G,L	CONNTYPE start option	APPN only; NODETYPE
		COSTBYTE	80		G,L	0	APPN only
		COSTTIME	81		G,L	0	APPN only
		CPCP	81		G,L	CPCP start option	APPN only
		CPNAME	81				
		DISCNT	83		G,L	(NO)	PU types 1, 2
		DYNADJCP	84		G,L	DYNADJCP start option	APPN only; CPNAME
		DYNLU	84		G,L	DYNLU start option	PU types 1, 2; CDRDYN
		ISTATUS	89		G	ACTIVE	
		LIMRES	89		G,L	NO	LU 6.2 only
		LMODADR	90				LPDATS; SEGMENT
		LPDA	92		G,L	ALLOW	
		LUGROUP	93				
		LUSEED	93				LUGROUP
		MAXDATA	95		G,L	261; 265	PU types 1, 2
		MAXOUT	96		G,L	1	
		NATIVE	99				BN=YES; APPN only
		NETID	100				PU types 1, 2
		NN	100		G,L		APPN only
		PASSLIM	102		G,L	value of MAXOUT	
		PDELAY	102		G,L	TERRESTR	APPN only
		PUTYPE	103		G,L	2	
		RETRIES	105		G,L	7	
		SECNET	106		G,L	NO	IBM 3710 Network Controllers, IBM 3174 token-ring attachments
		SECURITY	106		G,L	UNSECURE	APPN only; VM
		SEGMENT	107		G,L	1	
		SPAN	108				NetView
		STHRESH	109		G,L	(32768,32768, YES)	
		SUBAREA	110			*in_xid_value*	NETID; subarea nodes only
		TADDR	111			C1	PUTYPE=4
		TGN	111			ANY (APPN) 1 (subarea)	see description
		TGP	112		G,L		APPN only
		TOPO	112			CONNECT	APPN only
		UPARM1	113		G,L	128	APPN only
		UPARM2	113		G,L	128	APPN only

Table 8 (Page 3 of 3). Channel-Attachment (SDLC Nonswitched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		UPARM3	113		G,L	128	APPN only
		VERALSID	114		G,L	NO	APPN only
		XID	116		G,L	NO	PU types 1, 2
name	LU	LOCADDR	91	R			see description
		ASLENT	75		G,L,P		ASLTAB; PU types 1, 2
		ASLTAB	76		G,L,P		PU types 1, 2
		DLOGMOD	84		G,L,P		PU types 1, 2
		EAS	86		G,L,P	256	LOCADDR=0; PU types 1, 2
		ISTATUS	89		G	ACTIVE	
		LOGAPPL	91		G,L,P		PU types 1, 2; NetView
		LOGTAB	92		G,L,P		dependent LUs only
		MDLENT	96		G,L,P		MDLTAB; PU types 1, 2
		MDLTAB	97		G,L,P		PU types 1, 2
		MODETAB	98		G,L,P	ISTINCLM	PU types 1, 2
		PACING	101		G,L,P	(1,1,ADAP)	PU types 1, 2
		REGISTER	104		P	see description	APPN only
		SPAN	108				NetView
		SSCPFM	108		G,L,P	FSS	
		USSTAB	113		G,L,P	ISTINCDT	dependent LUs only
		VPACING	114		G,L,P	(2,1,ADAP)	PU types 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.

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.

SDLC Switched Lines

To define ICA support for SDLC switched lines, code the following definition statements:

```
VBUILD
GROUP LNCTL=SDLC, DIAL=YES
LINE
PU
```

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 68 for details on the VBUILD definition statement.

GROUP Definition Statement: Code a GROUP definition statement to define an SDLC switched line group.

Note that 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 SDLC switched line.

PU Definition Statement: Code one PU definition statement for each physical unit that can connect over the SDLC switched line.

Table 9 (Page 1 of 2). Channel-Attachment (SDLC Switched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	113	R			
		CONFGDS	79				VSE
		CONFGPW	79				VSE CONFGDS
name	GROUP	DIAL=YES	82	R			switched lines only
		LNCTL=SDLC	90	R			
		DIALNO	82				SHOLD; VSE
		DYNPU	85			NO	APPN only
		DYNPUPFX	86			CN	DYNPU=YES; APPN only
		SHOLD	108			NO	VSE ; X21SW; AUTODL; DIALNO
		SPAN	108				NetView
		SUBADIAL	110			NO	AUTO; CALL; AUTODL
		X21SW	116			NO	VSE X.21 short-hold mode
name	LINE	ACTIVTO	74		G	60	REPLYTO
		ADDRESS	74		G	030	
		ANSWER	75		G	ON	CALL=IN X21SW=NO
		AUTO	76				
		AUTODL	77		G	YES	VSE ; see description
		BUFCAP	77		G	1	
		CALL	78		G	IN	DIAL=YES; short-hold mode; applies only to network control mode
		ISTATUS	89		G	ACTIVE	
		LIMRES	89		G	NO	LU 6.2 only
		MAXBFRU	93		G	(2,8)	see description
		PAUSE	102		G	0.1	not valid for PU type 4
		REPLYTO	105		G	1.0	
		RETRIES	105		G	7	
		RETRYTO	106			12	VSE ; X21SW

Table 9 (Page 2 of 2). Channel-Attachment (SDLC Switched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		SERVLIM	107		G	4	see description
		SPAN	108				NetView
		SWNORDER	110		G	CPNAME	
		XMITDLY	116		G	2	type 2.1 connections
name	PU	AUTHLEN	76			YES	APPN only PU Type 2.1
		ISTATUS	89		G	ACTIVE	
		SPAN	108				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.

BSC Nonswitched Lines

To define ICA support for BSC nonswitched lines, code the following definition statements:

```
VBUILD
GROUP LNCTL=BSC
LINE
CLUSTER
TERMINAL
```

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 68 for details on the VBUILD definition statement.

GROUP Definition Statement: Code a GROUP definition statement to define a BSC line group.

Note that 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 BSC nonswitched line.

CLUSTER: Code one CLUSTER definition statement for each BSC 3270 cluster controller on the line.

TERMINAL: Code one TERMINAL definition statement for each BSC terminal attached to a 3270 cluster controller.

Table 10. Channel-Attachment (BSC Nonswitched) Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CA	113	R			
		CONFGDS	79				VSE
		CONFGPW	79				VSE CONFGDS
name	GROUP	LNCTL=BSC	90	R			
		SPAN	108				NetView
name	LINE	ADDRESS	74			030	
		ISTATUS	89		G	ACTIVE	
		RETRIES	105		G	7	
		SERVLIM	107		G	4	see description
		SPAN	108				NetView
name	CLUSTER	GPOLL	88	R			
		CUTYPE	81		G,L	3271	
		ISTATUS	89		G	ACTIVE	
		SPAN	108				NetView
name	TERMINAL	ADDR	74	R			
		TERM	111	R	G,L,C		
		ASLENT	75		G,L,C		ASLTAB; PU types 1, 2
		ASLTAB	76		G,L,C		PU types 1, 2
		DLOGMOD	84		G,L,C		PU types 1, 2
		FEATUR2	86		G,L,C	DUALCSE MODEL1 NOPRINTR NOSELPEN	
		ISTATUS	89		G	ACTIVE	
		LOGAPPL	91		G,L,C		PU types 1, 2; NetView
		LOGTAB	92		G,L,C	dependent LUs only	
		MDLENT	96		G,L,C		MDLTAB; PU types 1, 2
		MDLTAB	97		G,L,C		PU types 1, 2
		MODETAB	98		G,L,C	ISTINCLM	PU types 1, 2
		SPAN	108				NetView
		SSCPFM	108		C	USS3275	
		USSTAB	113		G,L,C	ISTINCDT	

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.

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

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.

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

Dynamic Change of Operands (VM, VSE): You can dynamically change the following SDLC nonswitched 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
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 “Dynamic Reconfiguration and Change of Operands” in the *VTAM Network Implementation Guide* and “VARY ACT Command” in *VTAM Operation* for more information on the VARY ACT,UPDATE=ALL command.

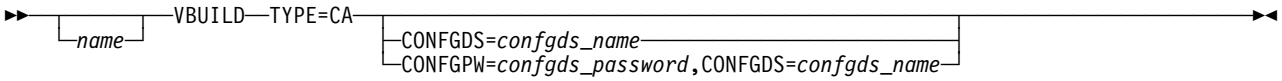
MODULO (VM, VSE): Note that when connecting an SDLC nonswitched ICA to an NCP, MODULO=8 must be coded on that NCP line. Currently, all hardware that supports SDLC nonswitched ICA has a setting of MODULO=8 and the hardware will not negotiate to another value. If MODULO=8 is not coded on the NCP line, an XID error occurs when the line is activated.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for a description of MODULO.

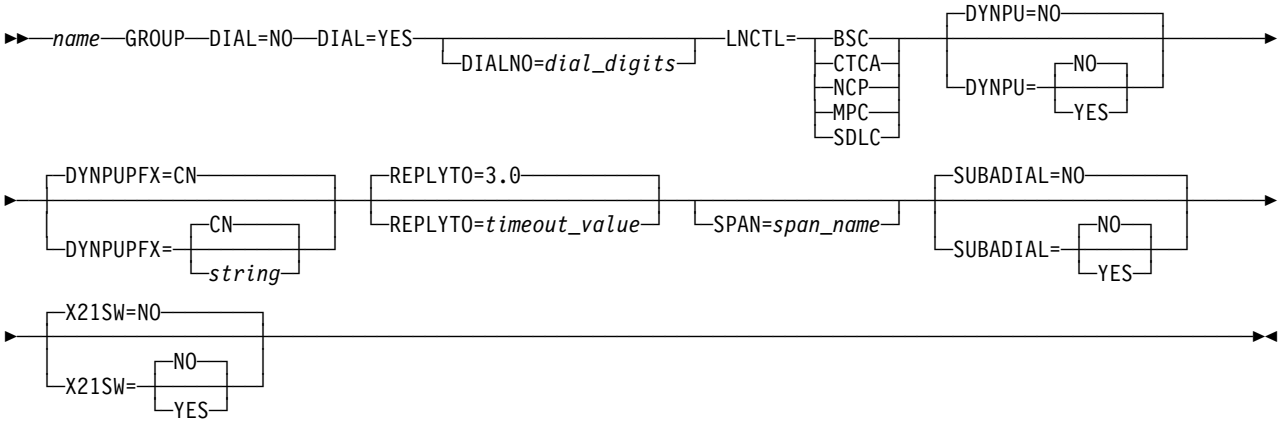
Full Syntax

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

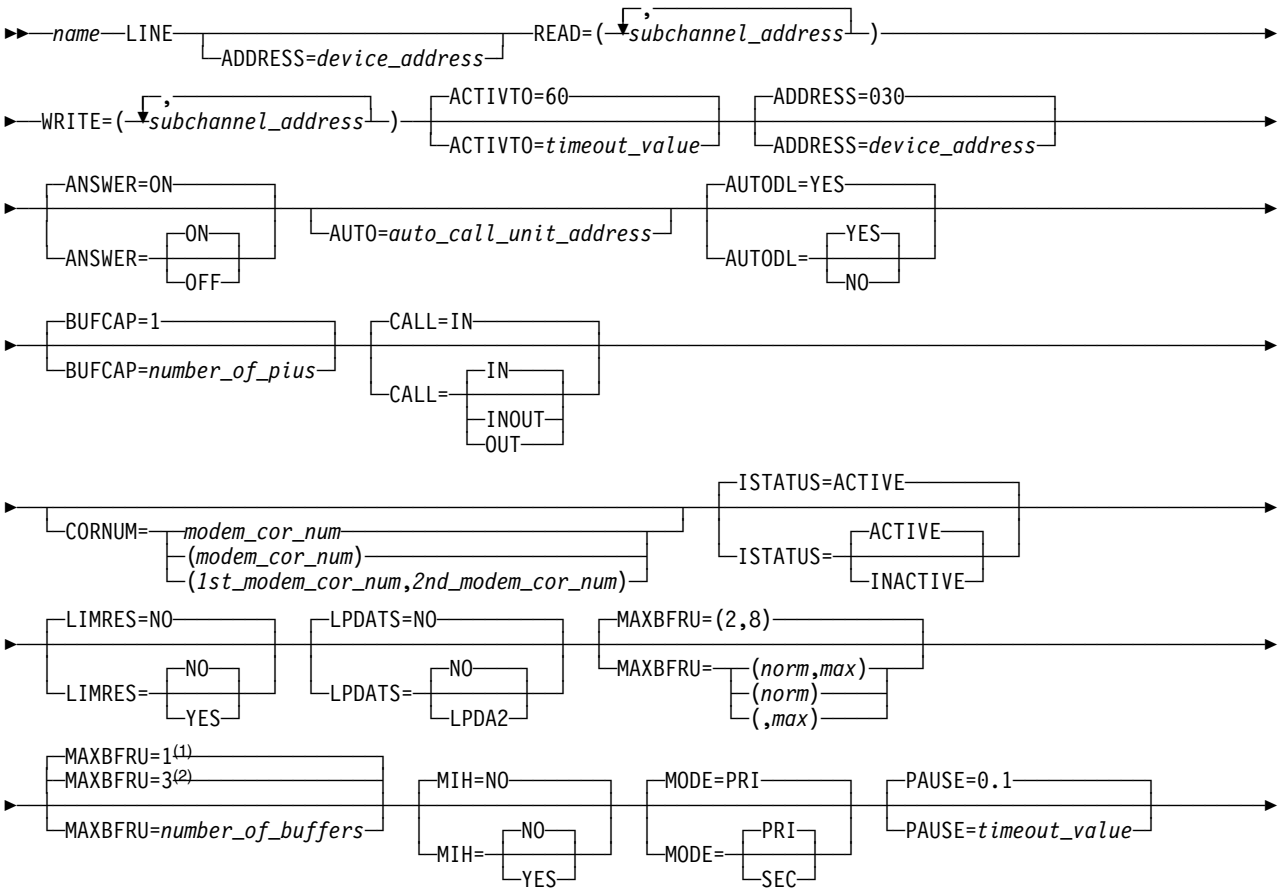
VBUILD

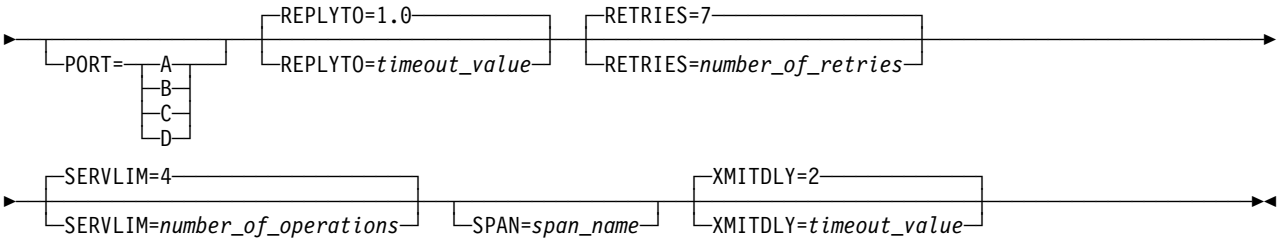


GROUP



LINE



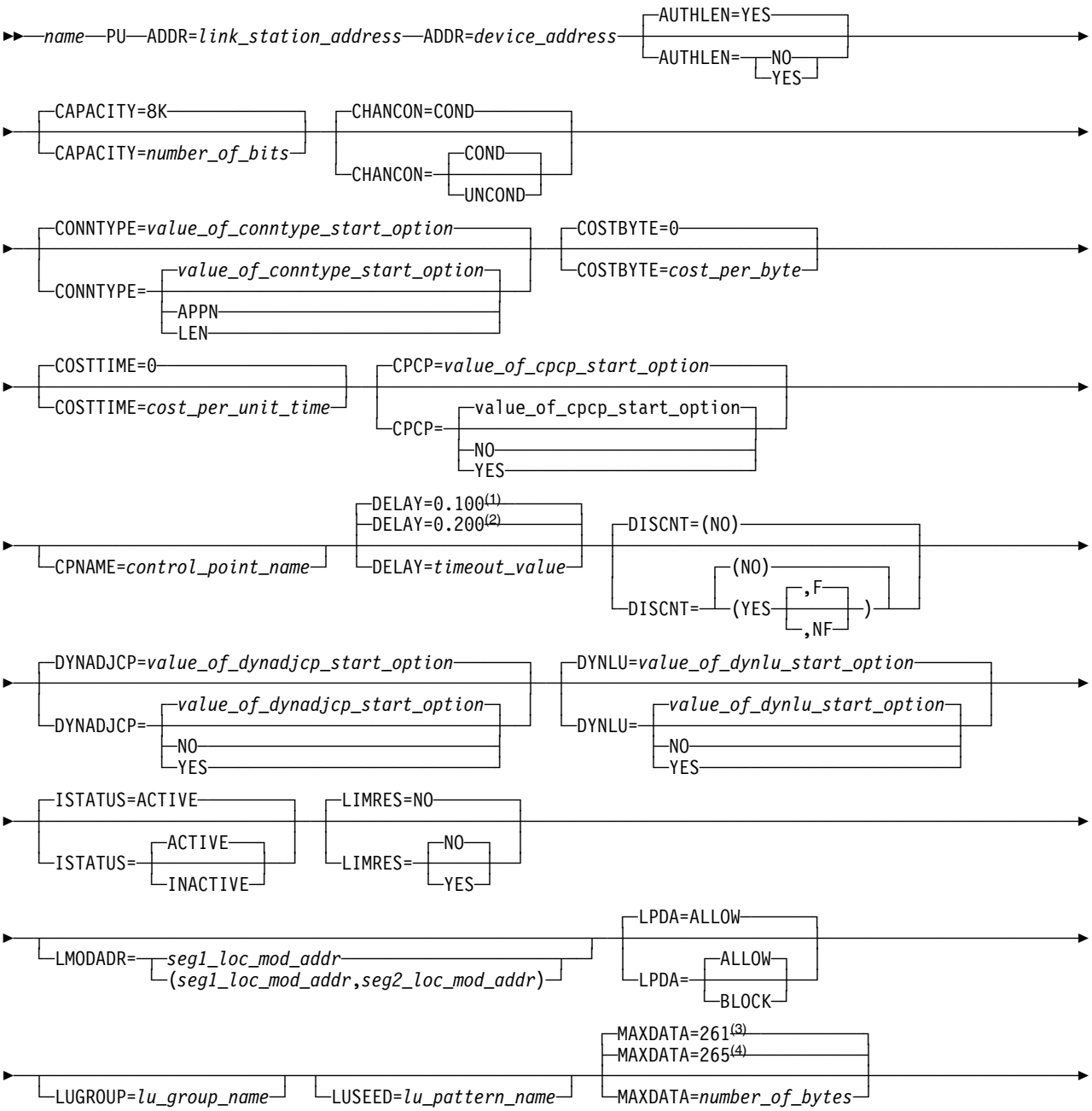


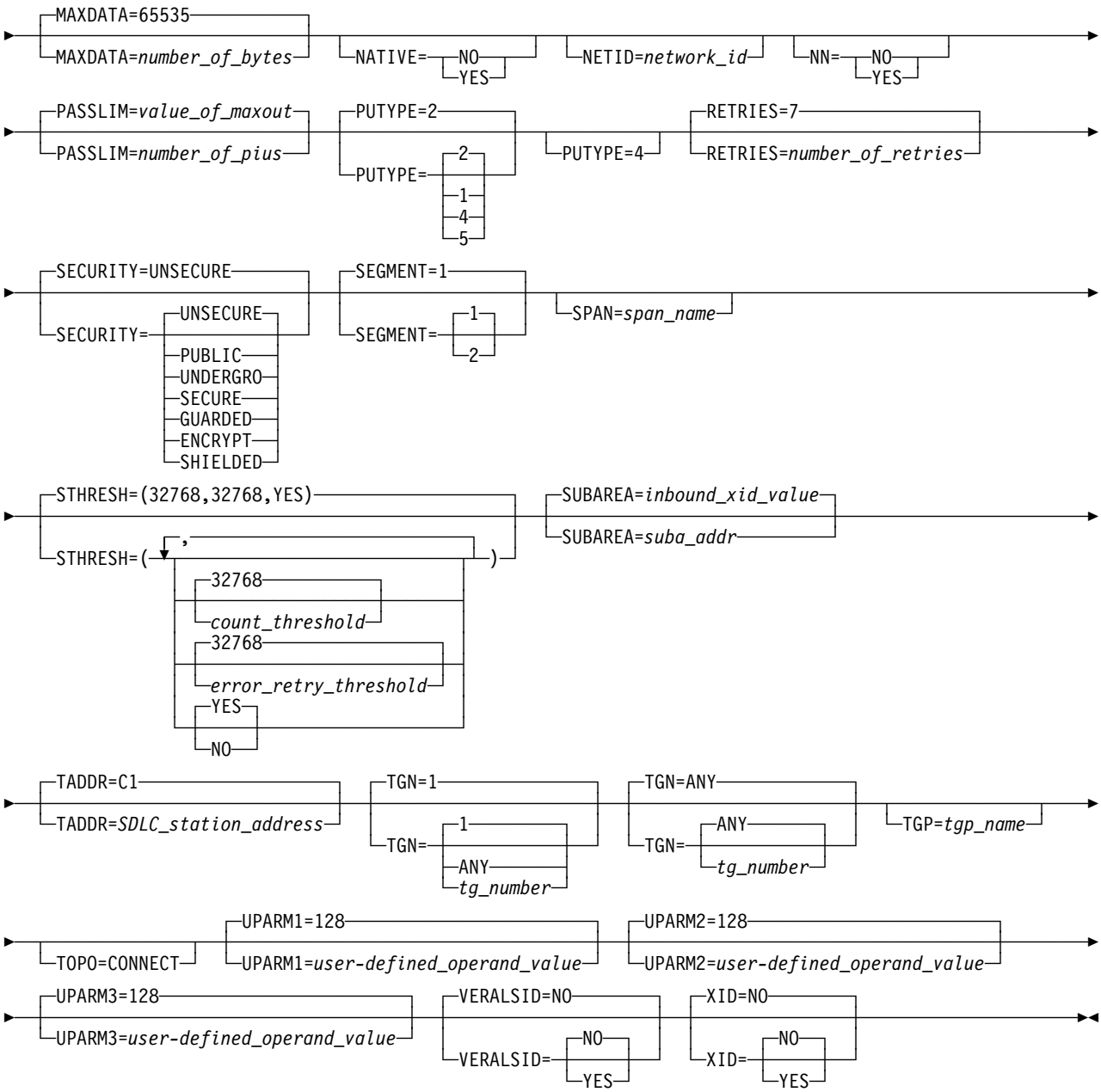
Notes:

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

[2] MAXBFRU defaults to 3 when LNCTL=NCP.

PU





Notes:

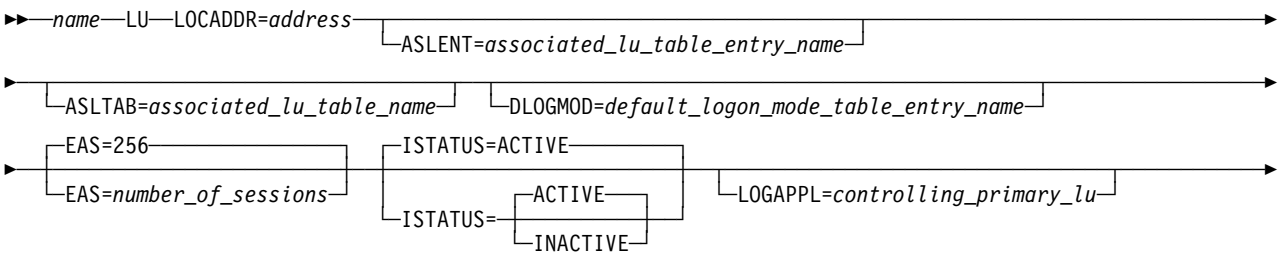
1 DELAY defaults to 0.100 when LNCTL=NCP is coded.

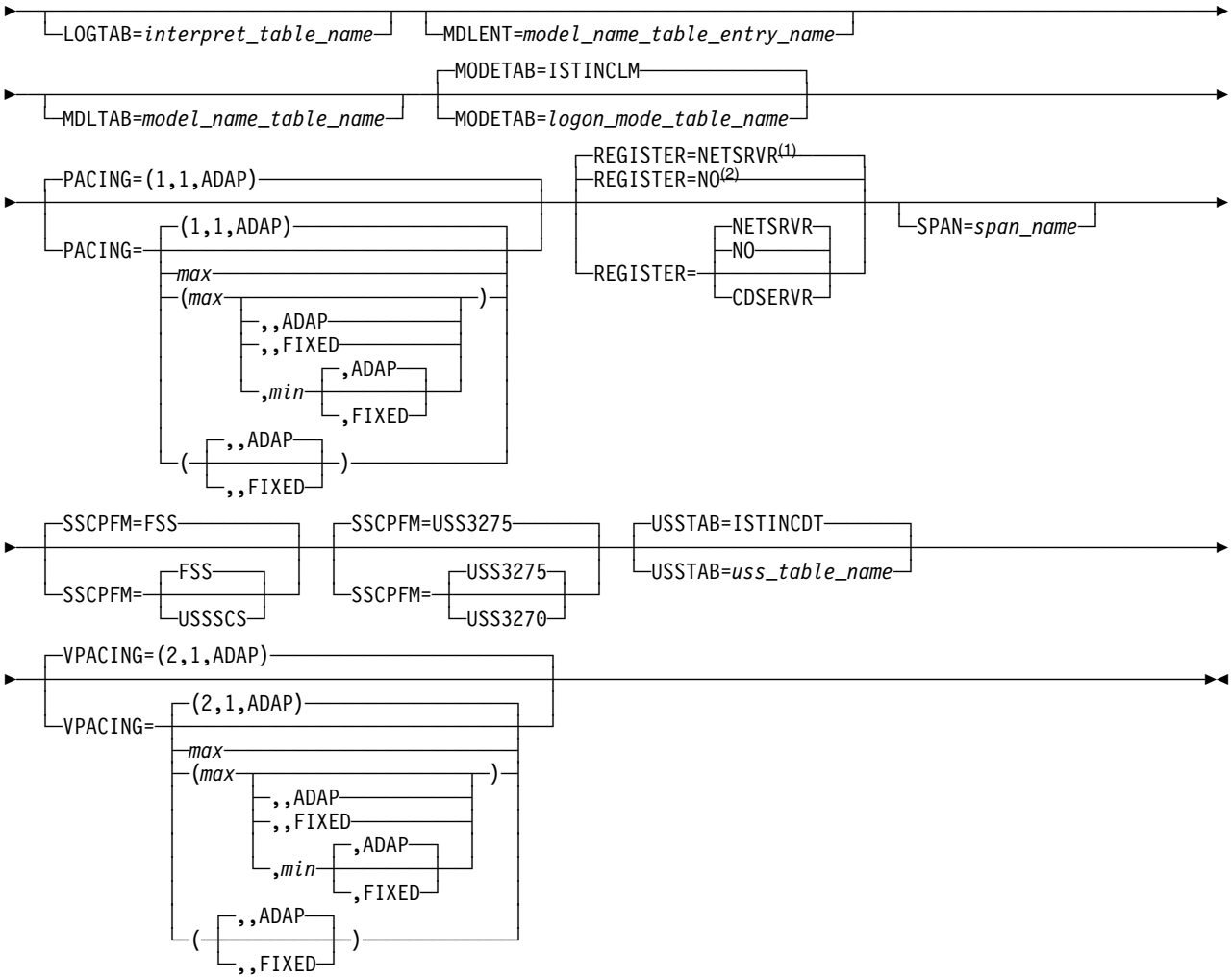
2 DELAY defaults to 0.200 when LNCTL=CTCA is coded.

3 MAXDATA defaults to 261 when PUTYPE=1 is coded.

4 MAXDATA defaults to 265 when PUTYPE=2 is coded.

LU



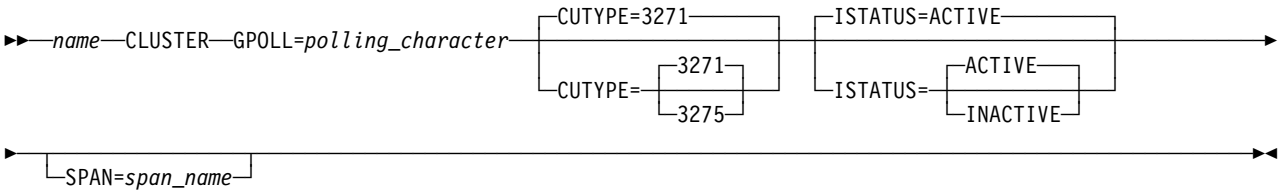


Notes:

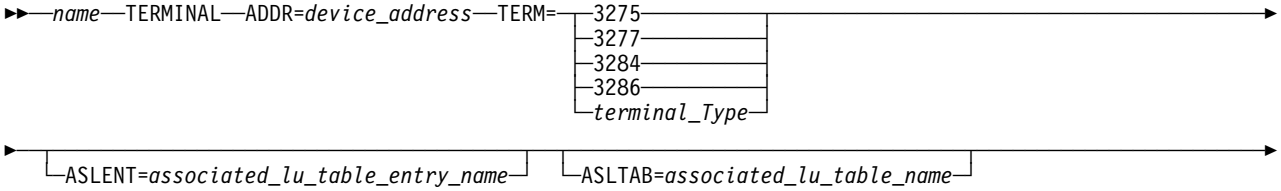
1 REGISTER defaults to NETSRVR for dependent LUs.

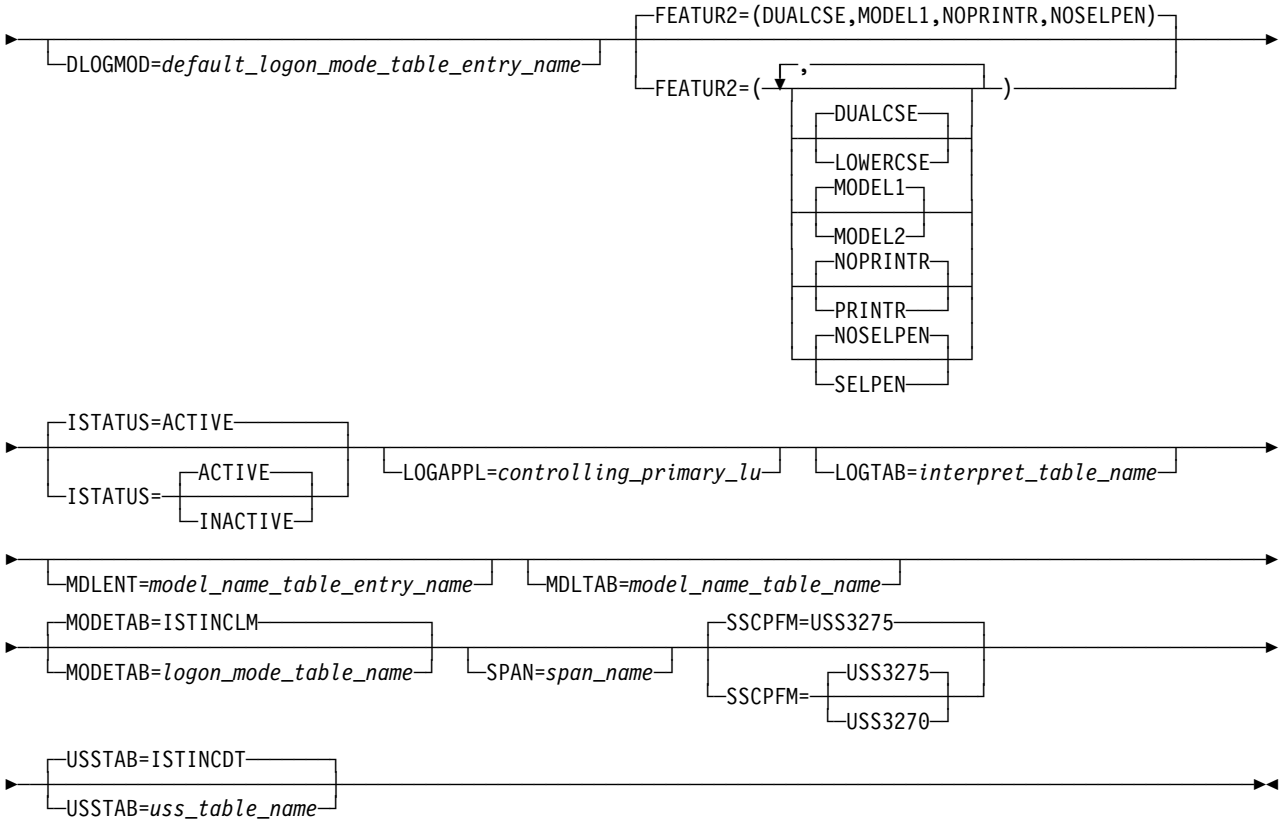
2 REGISTER defaults to NO for independent LUs.

CLUSTER



TERMINAL

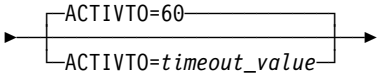



DLOGMOD=default_logon_mode_table_entry_name
FEATUR2=(DUALCSE,MODEL1,NOPRINTR,NOSELPEN)
FEATUR2=(
,
)
DUALCSE
LOWERCSE
MODEL1
MODEL2
NOPRINTR
PRINTR
NOSELPEN
SELPEN
ISTATUS=ACTIVE
ISTATUS=
ACTIVE
INACTIVE
LOGAPPL=controlling_primary_lu
LOGTAB=interpret_table_name
MDLENT=model_name_table_entry_name
MDLTAB=model_name_table_name
MODETAB=ISTINCLM
MODETAB=logon_mode_table_name
SPAN=span_name
SSCPFM=USS3275
SSCPFM=
USS3275
USS3270
USSTAB=ISTINCDT
USSTAB=uss_table_name

Operand Descriptions

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

ACTIVTO (VM, VSE)



statements:	GROUP, LINE
dependency:	REPLYTO
range:	1–255 seconds

specifies the interval, in seconds, that the integrated communication adapter waits without detecting an SDLC frame from another domain's NCP (PUTYPE=4) or from another VTAM with an integrated communication adapter (PUTYPE=5). After this interval, the integrated communication adapter signals a time-out error to VTAM (SDLC nonproductive receive time-out).

Note that the default is 60 seconds, unless you code REPLYTO. When you code REPLYTO, the default for ACTIVTO is 10 times the REPLYTO value.

ADDR

For SDLC

statements:	PU
dependency:	applies only to PU types 1, 2
range:	X'01'–X'FE'

For BSC

statements:	TERMINAL
dependency:	none

SDLC (VM, VSE): specifies the 2-digit hexadecimal SDLC station address for the physical unit. For instance, if the station address of a 3601 is X'B0', you code ADDR=B0 on the PU definition statement.

This address must be unique for each physical unit on the same line. Consult the relevant device publication for guidance in assigning this value.

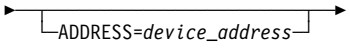
Note: This operand is required for PU types 1 and 2. If you code the ADDR operand for PU types 4 and 5, ADDR automatically defaults to C1.

BSC (VM, VSE): specifies the 2-digit hexadecimal representation of the EBCDIC device address (device selection character) assigned to this terminal. For example, if the terminal's device selection character is C, code ADDR=C3.

Refer to Table 11 on page 88 to find the right value for ADDR.

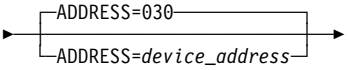
ADDRESS

For CTCA, NCP



statements:	LINE
dependency:	none

For SDLC, BSC



statements:	LINE
dependency:	none

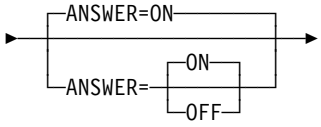
CTCA, NCP: specifies the three- or four-digit channel unit address of the channel-to-channel adapter (CTCA) or the channel-attached NCP. If you do not code this operand, the VTAM operator must supply the channel unit address by

specifying the U operand of the VARY ACT command. Refer to "VARY ACT Command" in *VTAM 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 channel unit address 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=*channel unit address*.) The second activation (either this definition or the automatic definition) will fail. For more information on channel-attached NCPs, refer to "Channel-Attached NCP" in the *VTAM Network Implementation Guide*.

SDLC, BSC: specifies the three-digit channel unit address for the SDLC nonswitched or switched line, or the BSC nonswitched line. The operator can override the defined address with the U operand of the VARY ACT command. See *VTAM Operation* for details on using the U operand.

ANSWER (VM, VSE)



statements: GROUP, LINE
dependency: CALL=IN

specifies whether physical units can dial in to VTAM.

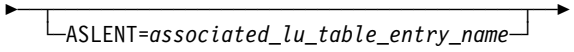
ANSWER=OFF
specifies that physical units cannot dial in to VTAM, regardless of whether the line is active or inactive.

ANSWER=ON
specifies that 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. See *VTAM Operation* for information on using the VARY ANS command.

Note: When CALL=OUT, the ANSWER operand is ignored.

ASLENT (VM, VSE)

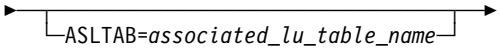


SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: ASLTAB; applies only to PU types 1 and 2

specifies the name of the associated LU table entry to be used for this logical unit (SDLC) or this terminal (BSC).

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 (VM, VSE)



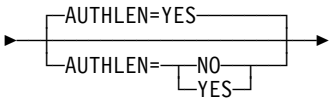
SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: applies only to PU types 1 and 2

specifies the name of the associated LU table to be used for the logical unit or the terminal. See “Associated LU Table” on page 638 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 714 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). See *VTAM Operation* for more information about the MODIFY TABLE command.

AUTHLEN (VM, VSE)



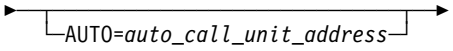
statements: PU
dependency: PU Type 2.1; APPN only

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.

AUTO (VM, VSE)



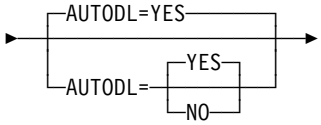
statements: LINE
dependency: X21SW=NO

specifies the address of an automatic calling unit (auto-call unit) for this line. The address of the auto-call unit is the same as the address of the line.

VSE This operand is valid only when X21SW=NO on the GROUP definition statement.

This operand is required if the line is a non-X.21 switched subarea connection and CALL=OUT or CALL=INOUT.

AUTODL (VSE)



statements: GROUP, LINE
dependency: see description

specifies whether an X.21 switched interface with the direct call facility accepts dial digits for outgoing calls:

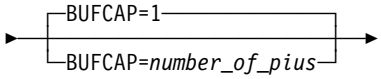
- If the direct call facility is provided on a per-call basis, code AUTODL=YES (the default).
- If the X.21 switched interface function is being used, code AUTODL=YES.
- If the direct call facility is provided on a subscription basis, code AUTODL=NO.

VTAM uses this information to avoid using a line that does not accept dial digits (AUTODL=NO) when a switched path statement is defined with dial digits.

This operand is valid only when X21SW=NO is specified on the GROUP definition statement.

You must code AUTODL=YES, if you coded SHOLD=(*free,npoll*) on the GROUP definition statement for this line or if you coded SUBADIAL=YES and X21SW=YES on the GROUP definition statement for this line.

BUFCAP (VM, VSE)



statements: GROUP, LINE
dependency: none
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 allocates is BUFCAP × MAXBFRU(MAX).

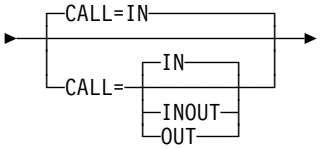
A retry value of 0 will not allow a type 2.0 or type 2.1 connection to become active. One retry is required to build the XID that will be exchanged.

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 tone 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 (VM, VSE)



statements: GROUP, LINE
dependency: DIAL=YES; short-hold mode;
applies only to network control mode

specifies whether physical units, or VTAM, or both, can initiate calls over a line.

CALL=IN
specifies that the line is to be used only for incoming calls (the physical units call VTAM).

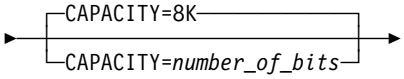
CALL=INOUT
specifies that the line is to be used for both incoming and outgoing calls.

CALL=OUT
specifies that the line is to be used only for outgoing calls (VTAM calls the physical units).

VSE If X.21 short-hold mode is activated, you must code CALL=INOUT.

This operand is valid only if DIAL=YES on the GROUP definition statement, and applies only to line operation in network control mode.

CAPACITY (VM, VSE)

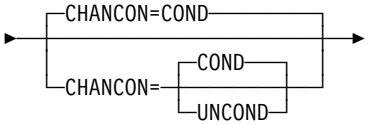


statements: GROUP, LINE, PU
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CHANCON



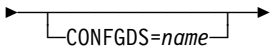
statements: GROUP, LINE, PU
dependency: LNCTL=NCP

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.

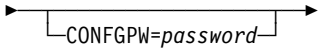
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

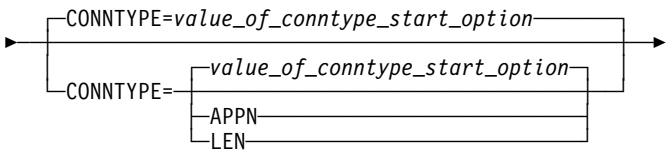


statements: VBUILD
dependency: CONFGDS

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 (VM, VSE)



statements: GROUP, LINE, PU
dependency: NODETYPE; APPN only

specifies for a PU 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.

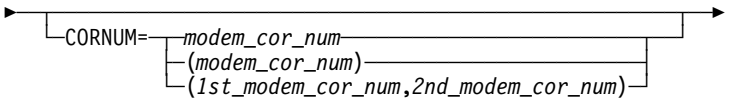
When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

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.

CORNUM (VM, VSE)



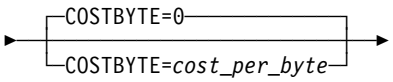
statements: LINE
dependency: none
range: 1–4095

specifies a correlation number for the line you are defining. You must specify the same value for each line attached to an IBM channelized modem.

You can specify a correlation number for the first DMPX modem and for a tailed DMPX modem, if any.

Because the network operator uses the correlation number to identify a specific DMPX modem on which a line is attached, each IBM channelized modem should have a unique correlation number. Therefore, *1st_modem_cor_num* and *2nd_modem_cor_num* should have different values. VTAM does not check correlation numbers.

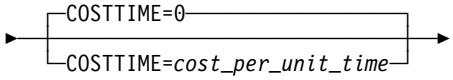
COSTBYTE (VM, VSE)



statements: GROUP, LINE, PU
dependency: 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.

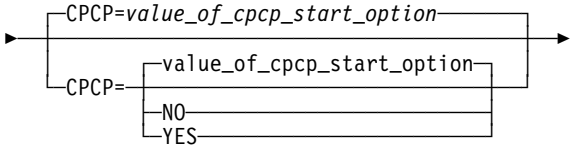
COSTTIME (VM, VSE)



statements: GROUP, LINE, PU
dependency: 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 (VM, VSE)



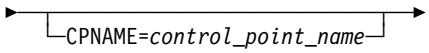
statements: GROUP, LINE, PU
dependency: APPN only

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.

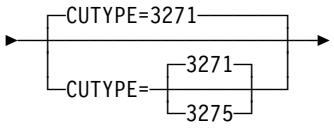
CPNAME (VM, VSE)



statements: PU
dependency: 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 in the form *netid.control_point_name*.

CUTYPE (VM, VSE)



statements: GROUP, LINE, CLUSTER
dependency: none

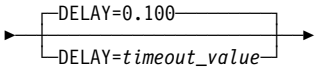
specifies whether the controller of the station is an IBM 3271 or 3275 BSC Controller.

CUTYPE=3271
specifies that the controller is an IBM BSC 3274, 3276, or 5937 controller.

CUTYPE=3275
specifies that the controller is an IBM BSC 3275 controller.

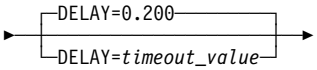
DELAY

For LNCTL=CTCA:



statements: GROUP, LINE, PU
dependency: none
range: 0-9.999 seconds

For LNCTL=NCP:



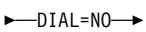
statements: GROUP, LINE, PU
dependency: none
range: 0-9.999 seconds

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 “Guidelines for Setting DELAY” in the *VTAM Network Implementation Guide* for more information.

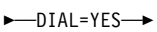
DIAL (VM, VSE)

For nonswitched lines



statements: GROUP
dependency: required for nonswitched lines

For switched lines

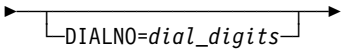


statements: GROUP
dependency: required for switched lines

SDLC nonswitched: specifies that the lines in the group do not require switched line control protocols.

SDLC switched: specifies that the lines in the group require switched line control protocols.

DIALNO (VSE)



statements: GROUP
dependency: required when SHOLD is coded

specifies the telephone number (in EBCDIC) of the port this GROUP definition statement represents. This number is used by the physical unit to re-establish the connection after it is broken by the X.21 short-hold mode/multiple-port-sharing feature. You can insert special characters in the number according to the following rules:

- You can insert as many vertical bars (“|”; X'FA') as you need to denote dialing pauses.
- For separator characters, use underscores (“_”; X'6D') or vertical bars (“|”; X'FA').
- Some modems require end-of-number characters. For end-of-number characters, use asterisks (“*”; X'5C'), percent signs (“%”; X'6C'), or at-signs (“@”; 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.

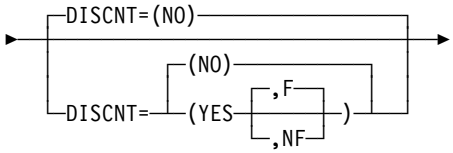
For auto-call (address call), X.21 switched lines always require a unique end-of-number character which must follow the dial digits. Do not code the required end-of-number character after the telephone number on the DIALNO operand because VTAM supplies it.

For X.21 lines, the *abbreviated address call* facility lets you use a shorter number instead of the full telephone number. Coding an abbreviated number is like coding a full number except that a period (X'4B') must precede the abbreviated number.

For example, suppose the common carrier facility defined the number 27 as the abbreviated number for the telephone number 555-1235. You could code DIALNO=.27 instead of DIALNO=5551235.

For X.21 lines, the *closed user group* facility lets you limit the number of physical units that can call or be called on a particular X.21 line. When you have defined the line to a switched physical unit as a member of a closed user group, that physical unit can call and be called only by other members of the group. For information on format requirements for closed user group numbers, contact your common carrier facility.

DISCNT (VM, VSE)



statements: GROUP, LINE, PU
dependency: applies only to PU types 1 and 2

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 “Switched SDLC Connection” in the *VTAM Network Implementation Guide* for a discussion of conditions that can cause sessions to terminate.

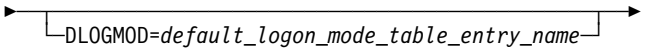
DISCNT=(YES)
If you code YES, VTAM automatically terminates the SSCP-LU and SSCP-PU sessions as soon as the last LU-LU session is terminated.

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 (VM, VSE)

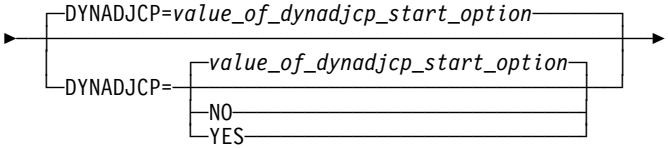


SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: applies only to PU types 1 and 2

specifies the name of the default logon mode table entry used by default if one is not 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 666.

DYNADJCP (VM, VSE)



statements: GROUP, LINE, PU
dependency: APPN only; ignored if CPNAME coded

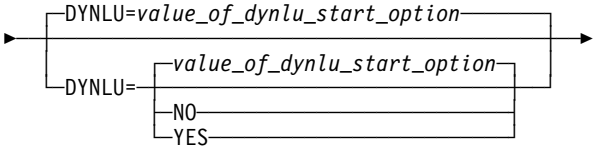
specifies whether an ADJCP can be created dynamically for an adjacent node.

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 (VM, VSE)



statements: GROUP, LINE, PU
dependency: CDRDYN; PU types 1, 2 only

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.

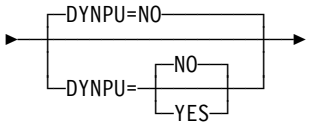
DYNLU=YES
specifies that dynamic allocation of CDRSC definitions is supported.

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 23 under the adjacent control point major node.

DYNPU (VM, VSE)



statements: GROUP (SDLC switched)
dependency: APPN only

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.

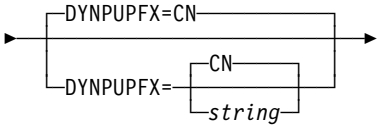
A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:

- 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 (VM, VSE)



statements: GROUP (SDLC switched)
dependency: DYNPU=YES; APPN only

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.

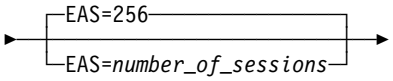
Note: This operand is valid only if DYNPU is specified as YES.

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 "Format of Definition Statements" on page 10 for naming conventions.

EAS (VM, VSE)



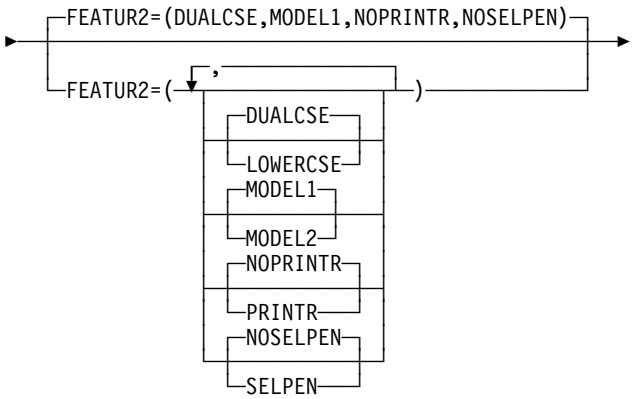
statements: GROUP, LINE, PU, LU
dependency: LOCADDR=0; applies only to PU types 1 and 2
range: 0–65535

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 534 for information about BSBUF.

FEATUR2 (VM, VSE)



statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: none

specifies the machine features for this 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.

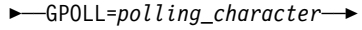
FEATUR2=MODEL1
FEATUR2=MODEL2
specifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

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 *VTAM Programming*.

FEATUR2=PRINTR
FEATUR2=NOPRINTR
specifies whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only when TERM=3275.

FEATUR2=SELPEN
FEATUR2=NOSELPEN
specifies whether this terminal supports a selector pen.

GPOLL (VM, VSE)



statements: CLUSTER
dependency: none

specifies the 2-digit hexadecimal representation of a single EBCDIC general polling character assigned to the cluster controller. For example, if the controller address is A, code GPOLL=C1.

Refer to Table 11 to find the right value for GPOLL.

Table 11 (Page 1 of 2). BSC 3270 General Polling (GPOLL) and Device Selection (ADDR) Characters

If CU or Device Number is	CU or Device Address is	Code GPOLL= or ADDR=
0	SP[1]	
1	A	C1
2	B	C2
3	C	C3
4	D	C4
5	E	C5
6	F	C6
7	G	C7
8	H	C8
9	I	C9
10	¢	4A
11	.	4B
12	<	4C
13	(	4D
14	+	4E
15	|	4F
16	&	50
17	J	D1
18	K	D2
19	L	D3
20	M	D4
21	N	D5
22	O	D6
23	P	D7
24	Q	D8
25	R	D9
26	!	5A
27	$	5B
28	*	5C

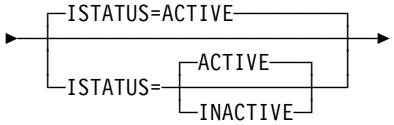
Table 11 (Page 2 of 2). BSC 3270 General Polling (GPOLL) and Device Selection (ADDR) Characters

If CU or Device Number is	CU or Device Address is	Code GPOLL= or ADDR=
29	)	5D
30	;	5E
31	¬	5F

Note:

1. Address SP (GPOLL or ADDR=40) is always used as the device address when CUTYPE=3275.

ISTATUS



statements: CLUSTER, GROUP, LINE, PU, TERMINAL
dependency: none

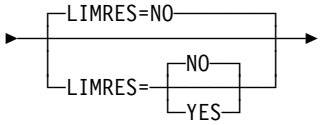
specifies whether the resource is to become active when its major node is activated.

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 "VARY ACT Command" in *VTAM Operation* for information on using the VARY ACT command.

LIMRES (VM, VSE)



statements: GROUP, LINE, PU
dependency: LU 6.2 only

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.

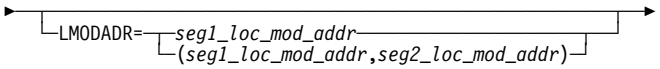
If you do not specify a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

LIMRES applies to LU 6.2 conversations only.

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.

LMODADR (VM, VSE)



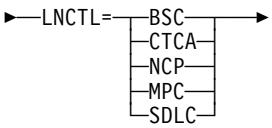
statements: PU
dependency: LPDATS; SEGMENT
range: X'0'–X'F'

specifies the local modem addresses used on the line going to the physical unit. LMODADR provides addresses to support LPDA-2 tests on multiple local modems. LMODADR is valid on the PU definition statement only if you specify LPDATS=LPDA2 on the LINE definition statement.

Specify a local modem address (*seg1_loc_mod_addr*) for link segment 1 and, if necessary, a local modem address (*seg2_loc_mod_addr*) for link segment 2. If SEGMENT=1, the default value is 01. If SEGMENT=2, the default value is (01,02).

Note: *seg2_loc_mod_addr* is valid only if SEGMENT=2.

LNCTL



statements: GROUP
dependency: none

specifies the type of line support being defined.

LNCTL=BSC
specifies that the lines are binary synchronous communication (BSC) lines.

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

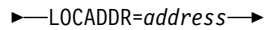
LNCTL=MPC
specifies that the lines are multipath channel-attachment lines.

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.

LNCTL=SDLC
specifies that the lines are synchronous data link control (SDLC) lines.

LOCADDR (VM, VSE)



statements: LU
dependency: see notes
range: 0–63 (PU type 1)
0–255 (PU type 2)

specifies the logical unit's local address at the physical unit.

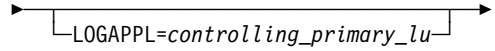
For independent logical units, specify 0. For dependent logical units, specify a non-zero value. All dependent logical units under a given physical unit must have unique LOCADDR values.

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 (VM, VSE)



SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: applies only to PU types 1 and 2; NetView

specifies 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.

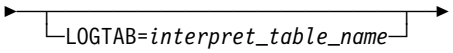
Automatic logon LU-LU sessions are not redriven after CP-CP outages even though the CP-CP sessions might redrive.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

LOGTAB (VM, VSE)

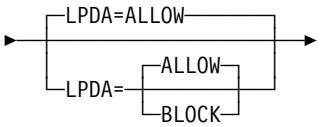


SDLC statements:	GROUP, LINE, PU, LU
BSC statements:	GROUP, LINE, CLUSTER, TERMINAL
dependency:	dependent LUs only

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 659 for a description of how to define an interpret table.

LOGTAB is valid only for dependent logical units.

LPDA (VM, VSE)



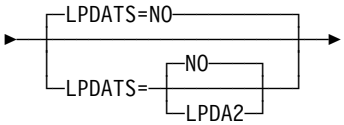
statements:	GROUP, LINE, PU
dependency:	none

specifies whether Link Problem Determination Aid (LPDA-2) tests can be run for the physical unit.

LPDA=ALLOW
specifies LPDA-2 tests are run for the PU.

LPDA=BLOCK
specifies no LPDA-2 tests are run for the PU.

LPDATS (VM, VSE)



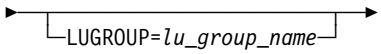
statements:	GROUP, LINE
dependency:	none

specifies whether the line uses modems that support LPDA-2 commands. LPDATS should be specified on a line that has the MODE operand specified as being primary.

LPDATS=LPDA2
specifies that the line uses modems that support LPDA-2 commands.

LPDATS=NO
specifies that the line does not use modems that support LPDA-2 commands.

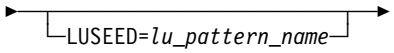
LUGROUP (VM, VSE)



statements: PU
dependency: 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 (VM, VSE)



statements: PU
dependency: 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:

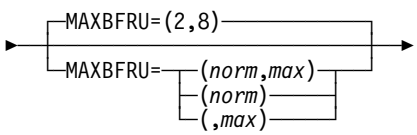
- The first character must be either alphabetical or one of the national characters (@ or $).
- The remaining characters can be either alphabetical, numerical, or national.
- There must be either two or three contiguous national # characters in the name. These characters represent holders, in which the 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 “Selection of Definitions for Dependent LUs (SDDLU) Exit Routine” in *VTAM Customization*.

This operand is valid only when you code the LUGROUP operand.

MAXBFRU

For SDLC



statements: GROUP, LINE
dependency: see description

specifies the number of buffers VTAM uses when it starts a normal channel program.

MAXBFRU=*norm*
specifies the number of buffers VTAM uses to read data from the line. The value of *norm* should be somewhat larger than that required for the average data transfer; if *norm* can hold the data, the path through VTAM is shorter than if more than *norm* but fewer than *max* buffers are required.

The default value for MAXBFRU is 2 for all types of physical units.

MAXBFRU=*max*
specifies the maximum number of buffers VTAM uses to read data from the line. The value for *max* must allocate enough buffers to hold the largest PIU that can be received over this line.

If you do not code *max*, the default value is 8 for all types of physical units.

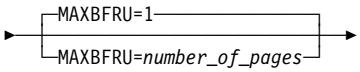
For PU type 2 and 2.1 lines, the largest PIU that can be contained is reduced by 20 bytes. To calculate the PIU capacity for these lines, use the following equation, using the *baseno* value for the IOBUF buffer pool:

```
PIU capacity = (IOBUF x MAXBFRU) - 20
```

For PU type 4 and 5 lines, the value should be large enough to hold the largest PIU that can be received times the number of PIUs that can be received on this line before a response is requested.

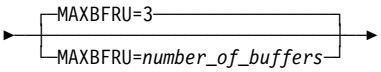
Some devices have a hardware restriction on the maximum number of buffers. VTAM will not check whether the value is in the correct range for the device type. Refer to the appropriate publication for the device before coding this operand.

For CTCA, MPC:



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

For NCP:



statements:	GROUP, LINE
dependency:	none
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.

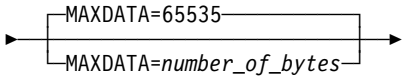
You can reduce this value after examining VTAM multipath channel tuning statistics so that excessive storage is not allocated for each subchannel. To obtain tuning statistics output, code the TNSTAT start option. Refer to Chapter 11, “Tuning VTAM for Your Environment” in the *VTAM Network Implementation 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.

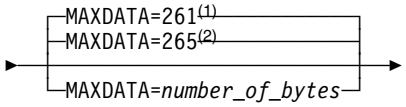
MAXDATA

For NCP



statements: GROUP, LINE, PU
dependency: LNCTL=NCP; applies only to PU type 4
range: 5–65535

For SDLC



Notes:

1 MAXDATA defaults to 261 when PUTYPE=1 is coded.

2 MAXDATA defaults to 265 when PUTYPE=2 is coded.

statements: GROUP, LINE, PU
dependency: LNCTL=SDLC; applies only to PU types 1 and 2
range: 5–65535

For NCP: 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 *VTAM Network Implementation Guide* for more information about choosing a value for MAXDATA.

For SDLC: 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/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.

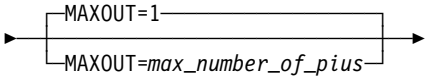
The default is 261 for a type 1 physical unit and 265 for a type 2 physical unit. The maximum amount of user data that VTAM sends to the physical unit in one PIU is:

- 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.

If you code this operand for a type 2.1 physical unit, it is overridden by the value derived from the format 3 XID received from the adjacent link station.

MAXOUT (VM, VSE)

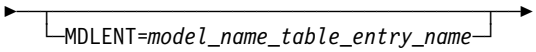


statements: GROUP, LINE, PU
dependency: none
range: 1–7

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.

MDLENT (VM, VSE)

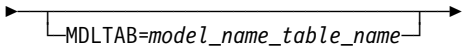


SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: MDLTAB; applies only to PU types 1 and 2

specifies the name of the model name table entry to be used for the logical unit or the terminal.

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 (VM, VSE)



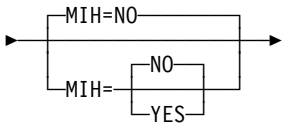
SDLC statements:	GROUP, LINE, PU, LU
BSC statements:	GROUP, LINE, CLUSTER, TERMINAL
dependency:	applies only to PU types 1 and 2

specifies the name of the model name table to be used for the logical unit or the terminal. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information on the MODIFY TABLE command.

MIH



statements:	GROUP,LINE
dependency:	LNCTL=CTCA

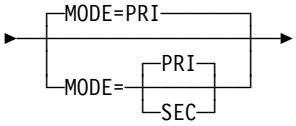
causes the link to become inoperative if a start I/O timeout occurs because the receiving host is in disabled compatibility mode. MIH stands for missing-interrupt handler. The default time value associated with this operand is 3 minutes, although you can change the default with the MIHTMOUT start option.

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.

MODE (VM, VSE)



statements: GROUP, LINE
dependency: none

specifies whether VTAM functions as a primary station or a secondary station.

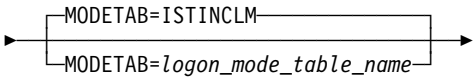
MODE=PRI
specifies that VTAM acts as a primary station. Specify PRI if there are only PU types 1 or 2 on the line.

MODE=SEC
specifies that VTAM acts as a secondary station. On a line with PU types 2.1 or 5, MODE can be either PRI or SEC. If the PU is a type 4, only SEC is accepted. In addition, if the PU is a type 4, the NCP cannot be activated or loaded over the line.

If you do not code this operand, and the physical unit is a PU type 5, whichever physical unit has the higher subarea value will be primary. Therefore, there is no default value for MODE for a PU type 5.

VM For 9371 and ES/9221 processors, solicited LPDA-2 commands are accepted only for a physical unit on a line defined with MODE=PRI. Unsolicited LPDA-2 commands are generated only for a line defined with MODE=PRI.

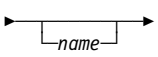
MODETAB (VM, VSE)



SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: applies only to PU types 1 and 2

specifies the name of a logon mode table to be used for the logical unit or the terminal. The name you code must be the name of a logon mode table created as described in “Logon Mode Table” on page 666. 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



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.

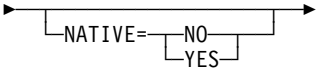
On LINE: *name* is required. It provides the minor node name of the line represented by this statement.

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.

NATIVE (VM, VSE)



statements: PU
dependency: 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, and

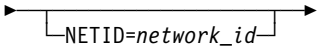
- if the NETIDs match, the connection defaults to a native connection
- if the NETIDs differ, the connection defaults to a nonnative connection.

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 (VM, VSE)



statements: PU
dependency: applies only to PU types 1 and 2

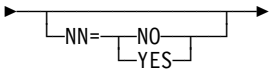
specifies a 1–8 character network name.

For type 2 physical units: This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See “XNETALS” on page 600 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, you must also omit SUBAREA. If you code NETID, and nonnative network connection is not supported, the definition is rejected unless the operand specifies VTAM's network ID.

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 names (NETIDs). Although VTAM allows you to use #, other products might enforce this restriction.

NN (VM, VSE)



statements: GROUP, LINE, PU
dependency: APPN only

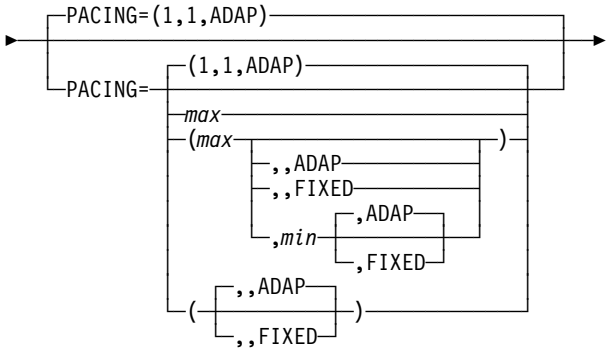
specifies whether the adjacent node is expected to be a network node.

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 (VM, VSE)



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 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.

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.

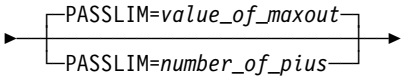
Defaults for PACING follow:

- 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 “Session Pacing” in the *VTAM Network Implementation Guide*.

PASSLIM (VM, VSE)

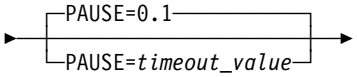


statements: GROUP, LINE, PU
dependency: none
range: 1–254

specifies the maximum number of contiguous PIUs or PIU segments that VTAM sends to the physical unit represented by this PU definition statement at one time.

If you do not code PASSLIM, or code it outside of its range, the value of MAXOUT is used as the default.

PAUSE (VM, VSE)

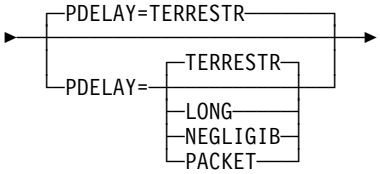


statements: GROUP, LINE
dependency: not valid for a line associated with a PU type 4
range: 0–25.5 seconds

specifies the interval that the integrated communication adapter waits, after completing a normal (data) poll cycle, before signaling completion. The integrated communication adapter sends the completion signal if all stations respond negatively (no data to send).

timeout_value is expressed in seconds and tenths of seconds.

PDELAY (VM, VSE)



statements: GROUP, LINE, PU
dependency: 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.

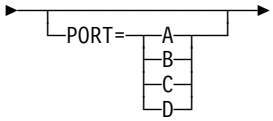
PDELAY=LONG
indicates a satellite delay (greater than 245.76 milliseconds).

PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

PDELAY=PACKET
indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=TERRESTR
indicates telephone network delay (between .48 and 49.152 milliseconds).

PORT (VM, VSE)



statements: LINE
dependency: none

specifies the port of the IBM channelized local modem to which the line is connected.

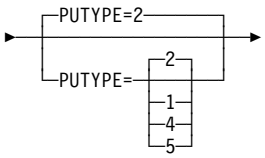
Note: Currently, only port A of an IBM channelized modem is LPDA-2-capable.

PUTYPE

For CTCA, MPC, NCP

statements: GROUP, LINE, PU
dependency: none

For SDLC



statements: GROUP, LINE, PU
dependency: none

CTCA, MPC, NCP: 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.

SDLC: specifies the physical unit type of the SDLC station represented by this PU definition statement:

PUTYPE=1
specifies an SNA terminal.

PUTYPE=2
specifies an SNA cluster controller or a type 2.1.

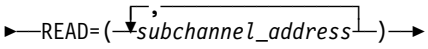
PUTYPE=4
specifies another domain's NCP in a communication controller.

PUTYPE=5
specifies another domain's VTAM and an integrated communication adapter (ICA).

VTAM allows stations of PU types 1 and 2 to be intermixed on the same multipoint line.

Code only one PU definition statement for each LINE definition statement when PUTYPE=4 or PUTYPE=5 is coded on the PU definition statement.

READ



statements: LINE
dependency: LNCTL=MPC
range: X'000'–X'FFFF' MVS
X'000'–X'FFF' VM,VSE

specifies the hexadecimal subchannel addresses (3 digits for VM and VSE; 3 to 4 digits for MVS) used to read data from the adjacent host. You must code at least one READ subchannel. If you code an address that is not defined to the system, the activation will fail for the entire group.

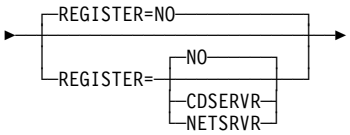
Subchannel addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen. VTAM generates addresses for the range. For example, READ=(132, 128-130, 13A) will define the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A. If any address in the range has not been defined to the system, the activation will fail.

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. If you do not code the corresponding address, the entire group will be deactivated.

Do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

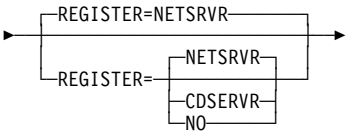
REGISTER

For Independent LUs:



statements: PU, LU
dependency: APPN only

For Dependent LUs:



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.

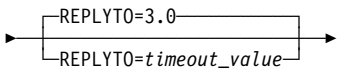
Dependent LUs are not usually the targets of LU-LU session setup attempts and are therefore not likely to be the targets of an APPN LOCATE search. However, because end nodes cannot be searched, you can register their dependent LUs with their network node server.

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

For independent LUs, NO is the default, and LOCADDR must be specified as 0.

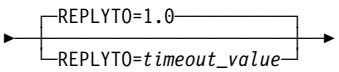
REPLYTO

For CTCA, MPC



statements: GROUP
dependency: MIH
range: 0.1–25.5 seconds

For SDLC



statements: GROUP, LINE
dependency: none
range: 0.1–25.5 seconds

For CTCA, MPC: 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.

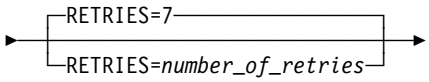
A STATION INOP can be scheduled at this time if a missing interrupt handler (MIH) purge has already taken place. The differences between the MIH and REPLYTO operands are:

- 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.

For SDLC: specifies the reply time-out value for the line when it operates as the primary station. If the integrated communication adapter has not received a response to a poll from the secondary station at the expiration of this interval, it notifies VTAM that a time-out error has occurred (idle detect timeout). VTAM retries the poll, select, or text-write up to the limit specified by RETRIES.

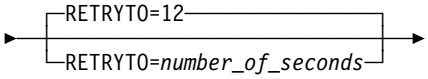
RETRIES (VM, VSE)



statements: GROUP, LINE
dependency: none
range: 0–255

specifies how many times VTAM tries to recover from errors that occur during transmission to or from the physical unit.

RETRYTO (VSE)

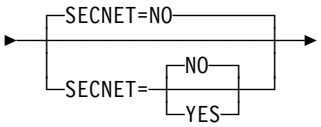


statements: LINE
dependency: valid only when X21SW=YES

specifies the number of seconds that the integrated communication adapter waits after receiving a call process signal that can be retried before it presents the signal to VTAM.

number_of_seconds can be 0, 1, 3, 6, 12, 18, or 24.

SECNET (VM, VSE)



statements: GROUP, LINE, PU
dependency: IBM 3710 Network Controllers or IBM 3174 token-ring attachments

specifies whether this physical unit is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

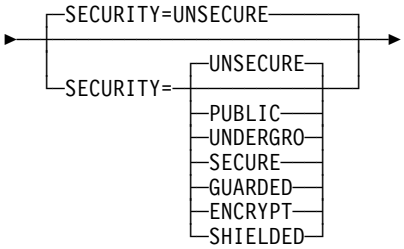
This operand applies only if there are IBM 3710 Network Controllers or IBM 3174 token-ring attachments in your network.

SECNET=NO
specifies that this physical unit is not associated with a secondary network.

SECNET=YES
specifies that this physical unit is associated with a 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 management application.

SECURITY (VM)



statements: GROUP, LINE, PU
dependency: 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.

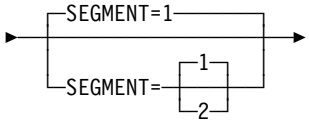
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

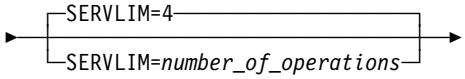
SEGMENT (VM, VSE)



statements: GROUP, LINE, PU
dependency: none

specifies the number of link segments between VTAM and the physical unit. Code 2 to specify that a second-level LPDA-2 circuit (a tailed link) exists for the physical unit you are defining.

SERVLIM (VM, VSE)



statements: GROUP, LINE
dependency: value specified is application-dependent
range: 0–255

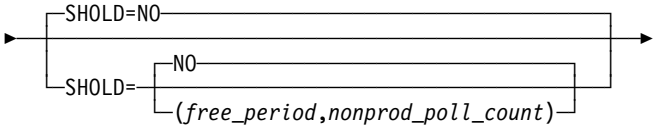
SDLC nonswitched: specifies the number of data-poll cycles to one contact-poll cycle, (that is, how many times each active physical unit on the line is asked to send data before attempting to contact additional physical units.)

For guidance on appropriate values, consult the relevant publication for the device.

SDLC switched: specifies the maximum number of output operations (WRITEs) that are allowed before a polling operation is started.

The value specified is application-dependent. When deciding what value to code, consider the amount of data sent with each output operation. As a guide, the value you code should allow a polling operation to occur to solicit operator input before subsequent output operations overwrite the 3270 screen. For guidance on appropriate values, consult the relevant publication for the device.

SHOLD (VSE)



statements: GROUP
dependency: X21SW; AUTODL; DIALNO
range: *free_period*: 0.1–3276.7
nonprod_poll_count: 1–32767

specifies whether this group is an X.21 short-hold mode/multiple port sharing (SHM/MPS) group.

SHOLD=NO
specifies that this group is not an SHM/MPS group.

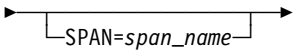
SHOLD=(*free_period,nonprod_poll_count***)**
specifies that this group is an SHM/MPS group and defines default values for the lines in the group. These values are:

free_period is the free period, in seconds, after dial completion.

nonprod_poll_count is the nonproductive polling count.

If you code SHOLD=(*free_period,nonprod_poll_count*), you must also code X21SW=YES, AUTODL=YES and DIALNO= *telephone number*. You may allow AUTODL to default to YES.

SPAN



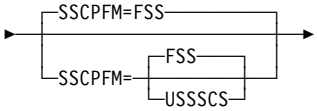
statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: NetView

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.

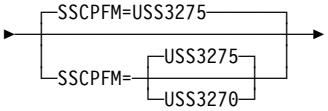
SSCPFM (VM, VSE)

For SDLC nonswitched:



statements: GROUP, LINE, PU, LU
dependency: none

For BSC nonswitched:



statements: CLUSTER, TERMINAL
dependency: 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.

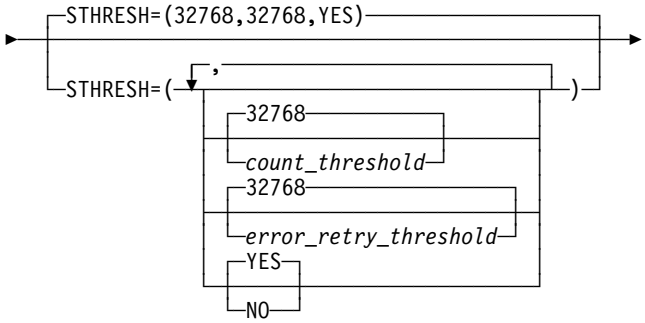
SSCPFM=USS3270
allows a user-defined logon message (USSMSG10) to print on BSC printers.

SSCPFM=USS3275
prevents a user-defined logon message (USSMSG10) from printing on BSC printers.

The SSCPFM operand has no effect on BSC terminals that are not 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.

STHRESH (VM, VSE)



statements: GROUP, LINE, PU
dependency: none

specifies the following values associated with this physical unit:

count_threshold
specifies the transmission count threshold for the total number of I-frames transmitted.

error_retry_threshold
specifies the transcription error retry threshold for the total number of retransmitted I-frames.

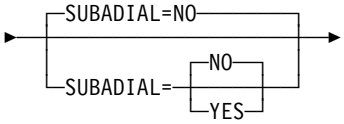
YES or NO
specifies whether VTAM is to perform LPDA-2 testing on the line when the transmission count threshold *m* has been reached. This value has no meaning if the physical unit is not LPDA-2-capable.

- YES means that VTAM tests the line and includes modem statistics in the network management information request generated for the NetView program.
- NO means that the test is not performed, but the network management information request is still generated for the NetView program.

When either of the thresholds specified by *count_threshold* or *error_retry_threshold* is reached, VTAM generates the following information for NetView:

- If the physical unit is LPDA-2-capable, the network management information request contains the result obtained from the unsolicited LPDA-2 request sent to the modem.
- If the physical unit is either not LPDA-2-capable or is on a nonswitched line that is not LPDA-2-capable, a network management information request is built without the modem data.
- If the physical unit is on a switched SDLC line, a record maintenance statistics (RECMS) record is built.

SUBADIAL (VM, VSE)



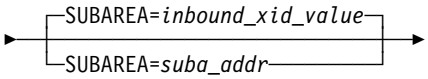
statements: GROUP
dependency: AUTO; CALL; AUTODL

specifies whether the lines in the group are used for switched subarea connections. If SUBADIAL=YES, each line in the group is a switched subarea line.

Note: For switched subarea connections, you must code AUTO when:

- CALL=OUT or CALL=INOUT, for non-X.21 lines
- AUTODL=YES, for X.21 lines (YES is the default)

SUBAREA (VM, VSE)

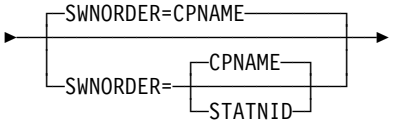


statements: PU
dependency: NETID; applies only to subarea nodes (PUTYPE=4 or 5)
range: 1–value of MXSUBNUM start option

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 received on the XID with the value you code on this operand. If the values do not match, the connection is terminated. If you do not code NETID, you should not code SUBAREA.

The value of SUBAREA must be the same as the subarea address defined for this physical unit in the other domains of the network.

SWNORDER



statements: GROUP, LINE
dependency: none

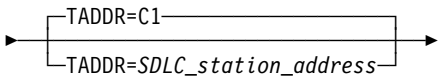
specifies the way VTAM locates a switched PU. If you do not code SWNORDER, VTAM searches for a PU by the CPNAME first. If VTAM does not find the PU by the CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

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.

TADDR (VM, VSE)

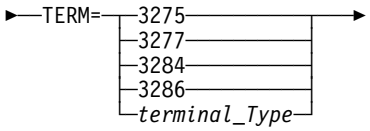


statements: PU
dependency: valid only when PUTYPE=4

specifies the 2-digit hexadecimal SDLC station address to use when VTAM is acting as a secondary station to a PU type 4 (an NCP). For example, if the station address is X'C0', you code TADDR=C0.

If the physical unit is connected to an NCP, the value of TADDR must match the value specified on the ADDR operand in the NCP LINE definition statements.

TERM (VM, VSE)

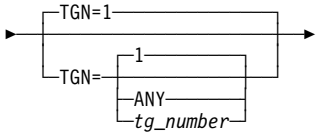


statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: none

specifies the type of device represented by this TERMINAL definition statement. Code 3277 for a BSC 3278; code 3286 for a 3287 or 3288.

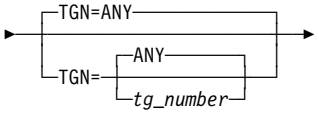
TGN

For subarea (CTCA, MPC, NCP, SDLC nonswitched):



statements: PU
dependency: PUTYPE=4,5
range: 1–255

For APPN (SDLC nonswitched):



statements: PU
dependency: PUTYPE=2, CPNAME, NETID
range: 0–21

Subarea (CTCA, MPC, NCP, SDLC nonswitched): 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.

APPN (SDLC nonswitched): 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 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.

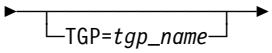
For APPN, coding TGN=0 means that the PU does not support parallel TGs.

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 (VM, VSE)

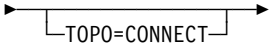


statements: GROUP, LINE, PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO (VM, VSE)



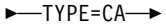
statements: PU
dependency: APPN only

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.

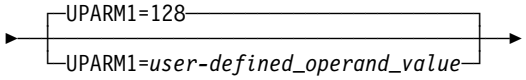
TYPE



statements: VBUILD
dependency: none

means that the VBUILD definition statement defines a channel-attachment major node to VTAM.

UPARM1 (VM, VSE)

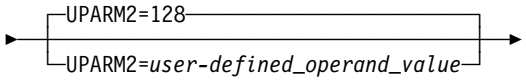


statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, "Forcing an APPN Route in a VTAM Network" in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

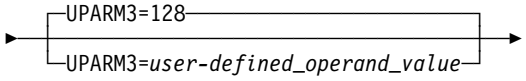
UPARM2 (VM, VSE)



statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

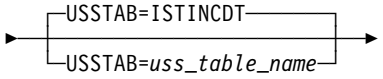
UPARM3 (VM, VSE)



statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

USSTAB (VM, VSE)

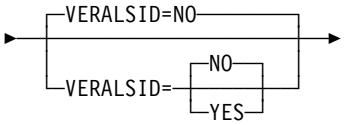


SDLC statements: GROUP, LINE, PU, LU
BSC statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: dependent LUs only;
applies only to PU types 1 and 2

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 690.

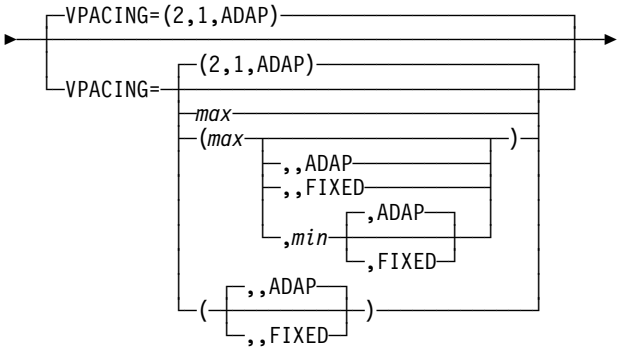
VERALSID (VM, VSE)



statements: GROUP, LINE, PU
dependency: 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 (VM, VSE)



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2
range: 0–63

specifies how VTAM paces the flow of data between the primary logical unit and VTAM's boundary function (2-stage pacing).

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).

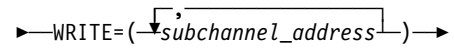
type is used only by NCP, and it requires at least NCP V5R4.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

WRITE



statements: LINE
dependency: LNCTL=MPC
range: X'000'–X'FFFF' **MVS**
X'000'–X'FFF' **VM,VSE**

specifies the hexadecimal subchannel addresses (3 digits for VM and VSE; 3 to 4 digits for MVS) used to write data to the adjacent host. If you specify an address that is offline or not defined to the system, the activation will fail for the entire group. You must code at least one WRITE subchannel address.

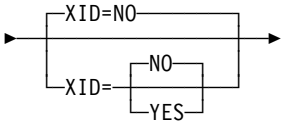
Subchannel addresses can be defined as a single address, as a range of addresses, or both. A range is indicated by a hyphen. VTAM generates addresses for the range. For example, WRITE=(132, 128-130, 13A) will define the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A. If any address in the range has not been defined to the system, the activation will fail.

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. If you do not code the corresponding address, the entire group will be deactivated.

Do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

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.

XID (VM, VSE)



statements: GROUP, LINE, PU
dependency: applies only to PU types 1 and 2

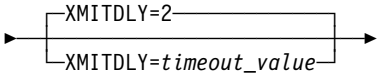
specifies how the physical unit is polled. XID is used primarily with type 2.1 peripheral nodes.

XID=NO
specifies that the physical unit is contact-polled with a Set Normal Response Mode (SNRM) command. If you specify NO, or use it by default, an exchange station identification (XID) is sent to the physical unit. For type 1 and 2 physical units, specify XID=NO.

XID=YES
specifies that polling is done by sending an initial XID, that is, a null XID. Specify YES for any physical unit that supports XID in normal disconnected response mode. For a type 2.1 physical unit, you must specify YES.

For more information about device-type dependencies, refer to *Planning for NetView, NCP, and VTAM*.

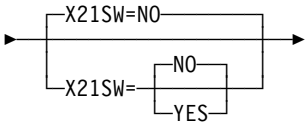
XMITDLY (VM, VSE)



statements: GROUP, LINE
dependency: applies only to type 2.1 connections
range: 1–255 seconds

specifies the number of seconds that VTAM delays its initial transmission after answering an incoming call.

X21SW (VSE)



statements: GROUP
dependency: X.21 short-hold mode

specifies whether the lines in the group are X.21 switched lines.

X21SW=NO
specifies that none of the lines in the group is an X.21 switched line.

X21SW=YES
specifies that all lines in the group are X.21 switched lines.

You must set this operand to YES if X.21 short-hold mode is activated.

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 *VTAM Network Implementation Guide* for detail 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” 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 12 on page 119. 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 12. 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

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.

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 MVS
  - LOGAPPL
  - LUALIAS[1]
  - 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:

[1] LUALIAS is only integrated in some cases. See Table 13 on page 122.

  - ALSREQ
  - CDRM
  - ALSLIST[2]
  - LUALIAS [1]
  - NETID
  - NQNMODE[3]
  - 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 119.)

  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 13 on page 122 for specific examples.

---

2 The list of adjacent link stations from the input resource and the existing resource are always combined.

3 NQNMODE is only integrated in some cases. See “NQNMODE Values after a Merge of Resource Definitions” on page 123.

Table 13. Examples of How LUALIAS Values Are Merged

Existing Resource	LUALIAS	Session	Input Resource	LUALIAS	Notes
Predefined Independent LU	N	No	Predefined CDRSC	Y	2
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.

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 14
- Table 15 on page 124
- Table 16 on page 125
- Table 17 on page 126.

Using the NQNMODE Merge Tables

- In Table 14, 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 15 on page 124 for input resources 1–5, or Table 16 on page 125 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 14 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 17 on page 126.

Table 14 (Page 1 of 2). 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
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

Table 14 (Page 2 of 2). 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
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 15 (Page 1 of 2). 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 17 on page 126.

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)
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)

Table 15 (Page 2 of 2). 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 17 on page 126.

EXIST. RES.	INPUT RESOURCE 1	INPUT RESOURCE 2	INPUT RESOURCE 3	INPUT RESOURCE 4	INPUT RESOURCE 5
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 16 (Page 1 of 2). 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 17 on page 126.

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)
13	**M**(2 4 12 13)	**F**(19)	**M**(1 5 13) **F**(20)	**F**(19)	**M**(1 22 13 6 17)

Table 16 (Page 2 of 2). 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 17.

EXIST. RES.	INPUT RESOURCE 6	INPUT RESOURCE 7	INPUT RESOURCE 8	INPUT RESOURCE 9	INPUT RESOURCE 10
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 17 (Page 1 of 2). Notes for NQNMODE Merges Tables. This table explains the letters and numbers seen in Table 15 on page 124 and Table 16 on page 125.

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.

Table 17 (Page 2 of 2). Notes for NQNMODE Merges Tables. This table explains the letters and numbers seen in Table 15 on page 124 and Table 16 on page 125.

Letter, Number	Explanation
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.
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.

Table 18. Cross-Domain Resource Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=CDRSC	143	R			
		CONFGDS	134				MVS,VSE
		CONFGPW	134				MVS,VSE CONFGDS
name	NETWORK	NETID	139				
name	GROUP	See CDRSC statement for operands.					
cdrscname	CDRSC	ADJLIST	130		G		
		ALSLIST	130		G		ALSREQ
		ALSREQ	131		G	ALSREQ start option	
		ASLENT	132		G		ASLTAB
		ASLTAB	132		G		
		CDRM	132		G		
		CPNAME	134		G		REGISTER
		DLOGMOD	134		G		MODETAB
		EAS	135		G	256	Independent LUs
		ISTATUS	135		G	ACTIVE	
		LOGAPPL	135		G		NetView or TSO (MVS)
		LUALIAS	136				
		MAXSESS	137		G		NCP-attached ILUs
		MDLENT	137		G		MDLTAB
		MDLTAB	138		G		
		MODETAB	138		G	ISTINCLM	
		NETSRVR	139		G		APPN only
		NQNMODE	139		G	NAME	see description
		PACING	141		G	1	
		REGISTER	141		G	NO	CPNAME; APPN only
		RESSCB	142		G	0	PU Type 2.1
		SPAN	142		G		NetView
		SRCOUNT	142		G		SRCHRED start option
		SRTIMER	143		G		SRCHRED start option
		VFYOWNER	143		G	NO	
		VPACING	144		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.

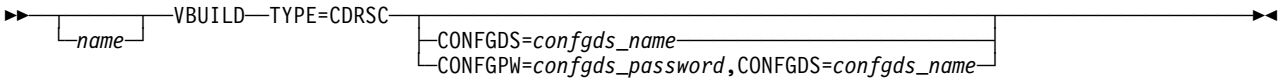
Notes:

1. MVS Information on coding the CDRSC definition statement for TSO/VTAM is in "Multiple-Domain Network" in the *VTAM Network Implementation Guide*.
2. The same set of CDRSC definition statements can be used throughout the network.

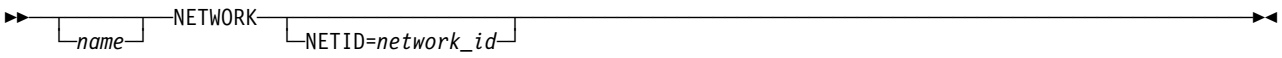
Full Syntax

The full syntax for the cross-domain resource (CDRSC) major node follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

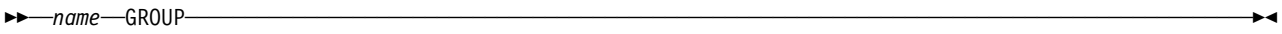
VBUILD



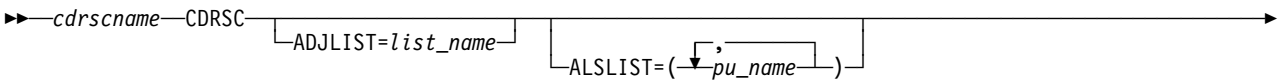
NETWORK

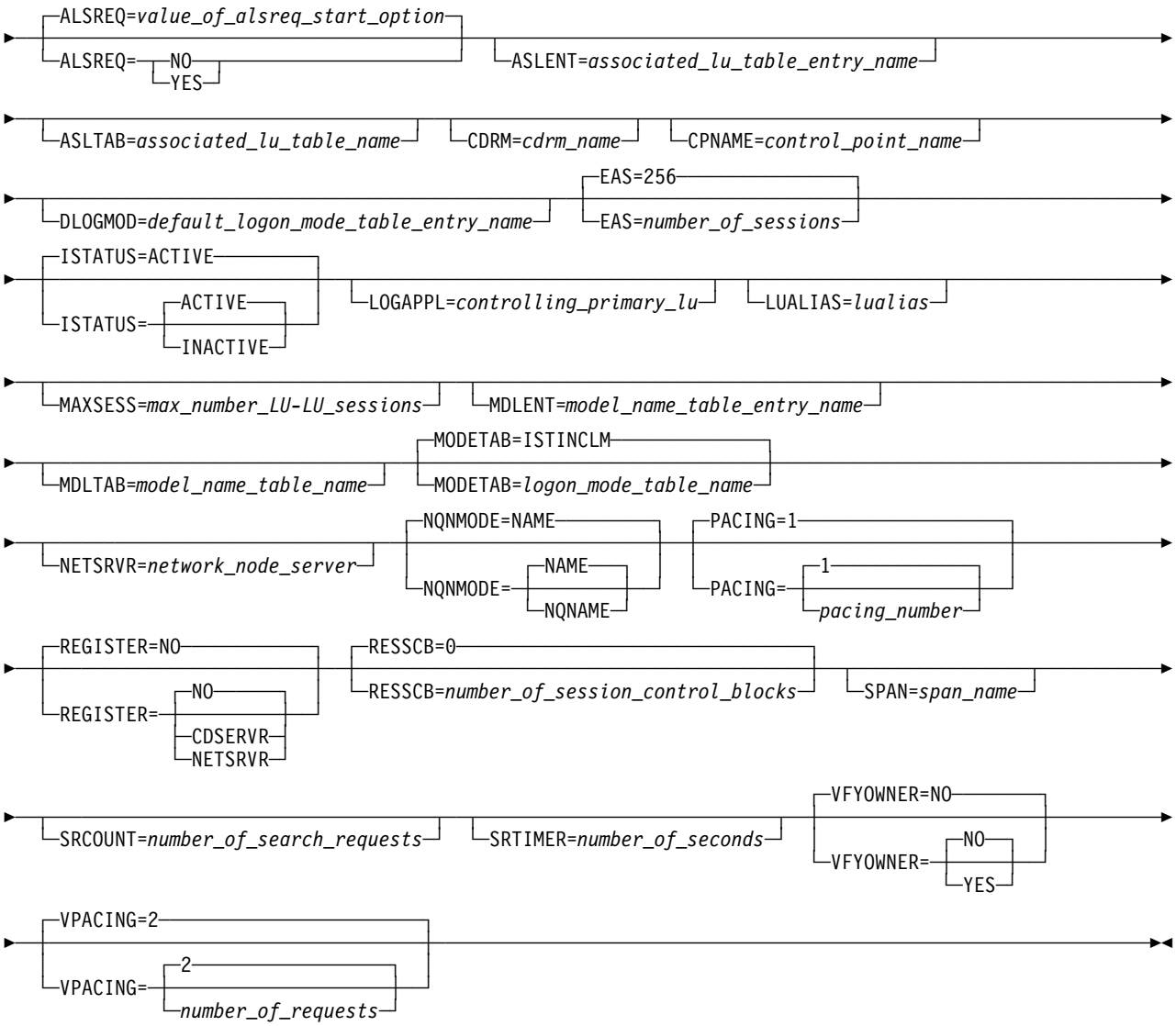


GROUP



CDRSC

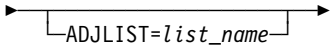




Operand Descriptions

The following section describes the operands you can code for the cross-domain resource (CDRSC) major node.

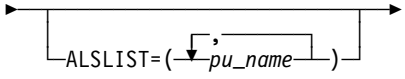
ADJLIST



statements: GROUP, CDRSC
dependency: 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



statements: GROUP, CDRSC
dependency: 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.

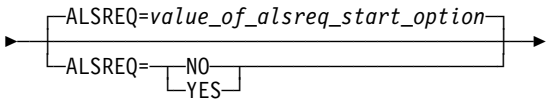
You can include ISTAPNPU in the list instead of a real adjacent link station name. ISTAPNPU is a generic representation for an APPN adjacent link station. If VTAM finds ISTAPNPU in the adjacent link station list for an independent LU, VTAM treats ISTAPNPU as if it were a real APPN adjacent link station 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 “Dynamic Selection of Session Connections for LEN-Attached Independent LUs” in the *VTAM Migration Guide for MVS/ESA* for details.

ALSREQ



statements: GROUP, CDRSC
dependency: 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.

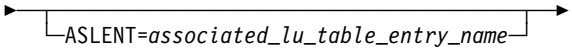
ALSREQ=YES
specifies that an adjacent link station name must match an entry in the default adjacent link station list of the independent LU at a session request.

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

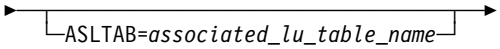


statements: GROUP, CDRSC
dependency: ASLTAB

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



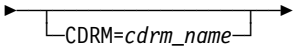
statements: GROUP, CDRSC
dependency: none

specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 638 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

CDRM



statements: GROUP, CDRSC
dependency: 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.

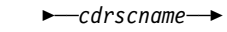
If you coded NETID on the NETWORK definition statement, thereby associating it with the CDRSC, and the CDRM name is not the SSCP name of the CDRM in which the resource resides, the session setup could fail. See "VFYOWNER" on page 143 for more details.

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 "MODIFY CDRM Command" in *VTAM Operation* for more information about the MODIFY CDRM command.

cdrscname



statements: CDRSC
dependency: none

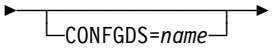
specifies the name of a resource (a logical unit or application program) in another domain or network that this CDRSC is to represent.

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.

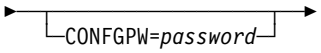
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

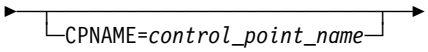


statements: VBUILD
dependency: CONFGDS

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



statements: GROUP, CDRSC
dependency: REGISTER=NO

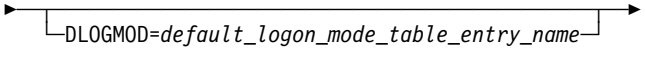
identifies the resource's owning control point name. To allow definition of casually connected resources, this operand can be fully qualified with a network ID. Although it is not recommended, if you define an APPN resource to avoid APPN broadcasts to locate it, then CPNAME is required.

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



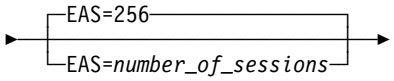
statements: GROUP, CDRSC
dependency: MODETAB

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 666.

EAS



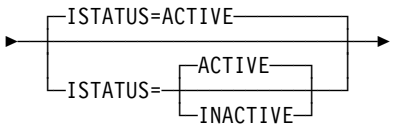
statements: GROUP, CDRSC
dependency: valid only for ILUs (LOCADDR=0)
range: 0–65535

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 534 for information about BSBUF.

ISTATUS



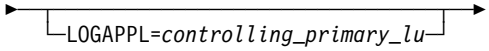
statements: GROUP, CDRSC
dependency: none

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



statements: GROUP, CDRSC
dependency: NetView or TSO (MVS)

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.

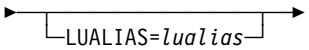
To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

LUALIAS



statements: CDRSC
dependency: 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:

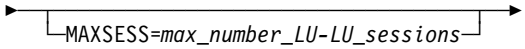
- When the LUALIAS name has been defined for a resource that is defined on the CDRSC definition statement that is preceded by a NETWORK definition statement.
- For session requests that cross the application program interface
- For session requests whose origin or destination is the VTAM for which LUALIAS 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.

MAXSESS

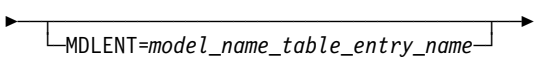


statements: GROUP, CDRSC
dependency: Independent LUs attached to an NCP
range: 1–65535

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

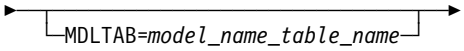


statements: GROUP, CDRSC
dependency: MDLTAB

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



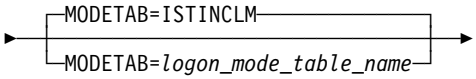
statements: GROUP, CDRSC
dependency: none

specifies the name of the model name table to be used for the logical unit. See "Model Name Table" on page 683 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 714 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 "MODIFY TABLE Command" in *VTAM Operation* for more information about the MODIFY TABLE command.

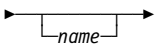
MODETAB



statements: GROUP, CDRSC
dependency: 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 666. 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



statements: VBUILD, NETWORK

statements: GROUP

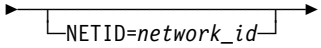
On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

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



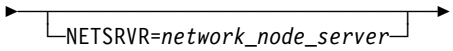
statements: NETWORK
dependency: none

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



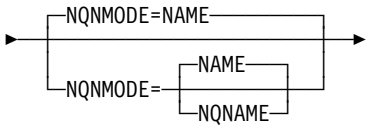
statements: GROUP, CDRSC
dependency: APPN only

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



statements: GROUP, CDRSC
dependency: 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.

Note: 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.

If 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 19 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.

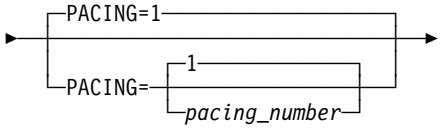
Table 19 (Page 1 of 2). How NQNMODE Relates to CDRSCs

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

Table 19 (Page 2 of 2). How NQNMODE Relates to CDRSCs

Type of CDRSC	CDRSC or Start Option	NQNMODE=NAME	NQNMODE=NQNAME
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



statements: GROUP, CDRSC
dependency: 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.

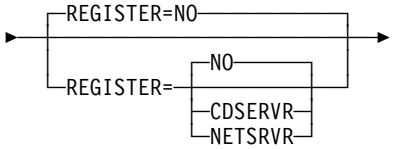
Notes:

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

REGISTER



statements: GROUP, CDRSC
dependency: CPNAME; APPN only

specifies how a resource should be registered on a definition statement.

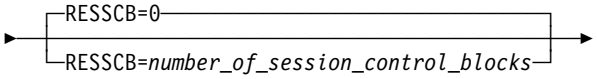
Note that for the cross-domain resource major node, CPNAME and REGISTER are conflicting options. If you code CPNAME and a REGISTER value of anything other than NO, an error message is issued and REGISTER is set to NO.

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



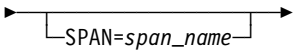
statements: GROUP, CDRSC
dependency: valid only for PU type 2.1
range: 0–65535

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

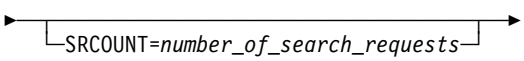


statements: GROUP, CDRSC
dependency: NetView

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



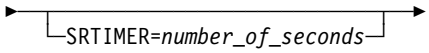
statements: GROUP, CDRSC
range: 0–65535
dependency: SRCHRED start option

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



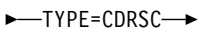
statements: GROUP, CDRSC
range: 0–65535
dependency: SRCHRED start option

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.

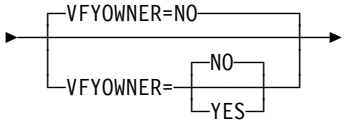
TYPE



statements: VBUILD
dependency: none

indicates a CDRSC major node. This operand is required.

VFYOWNER



statements: GROUP, CDRSC
dependency: none

specifies whether owner verification takes place when sessions are attempted with this resource.

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:

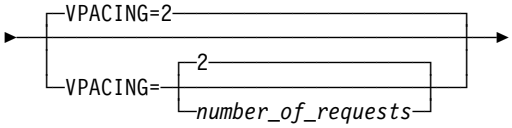
- The CDRSC is predefined with the NETID operand on the NETWORK definition statement.
- You code CDRM=*cdrmname*.

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 the two conditions listed under VFYOWNER=YES are met:

- The CDRSC is predefined with the NETID operand on the NETWORK definition statement.
- CDRM=*cdrmname* is coded.

Note: The VFYOWNER function is supported only in subarea sessions.

VPACING



statements: GROUP, CDRSC
dependency: 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 “Session Pacing” in the *VTAM 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 20 (Page 1 of 2). 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	160	R			
		CONFGDS	152				MVS,VSE
		CONFGPW	152				MVS,VSE CONFGDS
name	NETWORK	NETID	156				
cdrmname	CDRM	SUBAREA	159	R			
		CAPACITY	150			8K	VRTG=YES; APPN only
		CDRDYN	151			NO	
		CDRSC	152			REQ	SSCP-SSCP sessions
		COSTBYTE	153			0	VRTG=YES; APPN only
		COSTTIME	153			0	VRTG=YES; APPN only
		DISJOINT	153			NO	APPN only
		ELEMENT	154			1	
		GWSELECT	154			YES	cross-net CDRMs
		ISTATUS	155			ACTIVE	
		NN	156				VRTG=YES; APPN only
		PDELAY	157			TERRESTR	VRTG=YES; APPN only
		RECOVERY	157			YES	
		SECURITY	158			UNSECURE	VRTG=YES; APPN only
		SPAN	158				MVS,VM NetView
		TGP	159				VRTG=YES; APPN only
		UPARM1	160			128	VRTG=YES; APPN only
		UPARM2	160			128	VRTG=YES; APPN only
		UPARM3	160			128	VRTG=YES; APPN only
		VPACING	160			63	
		VRTG	161			VRTG start option	APPN only; NODETYPE; HOSTSA
		VRTGCPCP	161			VRTGCPCP start option	VRTG=YES; APPN only; NODETYPE; HOSTSA
name	GWPATH	ADJNET	149			NETWORK	MVS,VM
		ADJNETCS	149			ISTVTCOS	
		ADJNETEL	150			1	ADJNET
		ADJNETSA	150				
		ELEMENT	154			1	
		GWN	154				PUNAME on BUILD
		SUBAREA	159				

Table 20 (Page 2 of 2). Cross-Domain Resource Manager Major Node Definition Statement and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
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: MVS,VM 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.

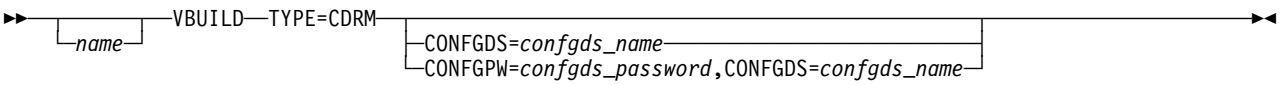
One or more GWPATH definition statements can follow a CDRM definition statement for a CDRM in another network. The order of the GWPATH definition statements is the order VTAM uses when selecting the path for a given session.

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.

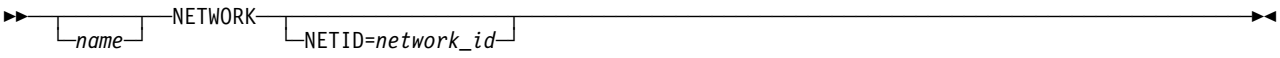
Full Syntax

The full syntax for the cross-domain manager (CDRM) major node follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

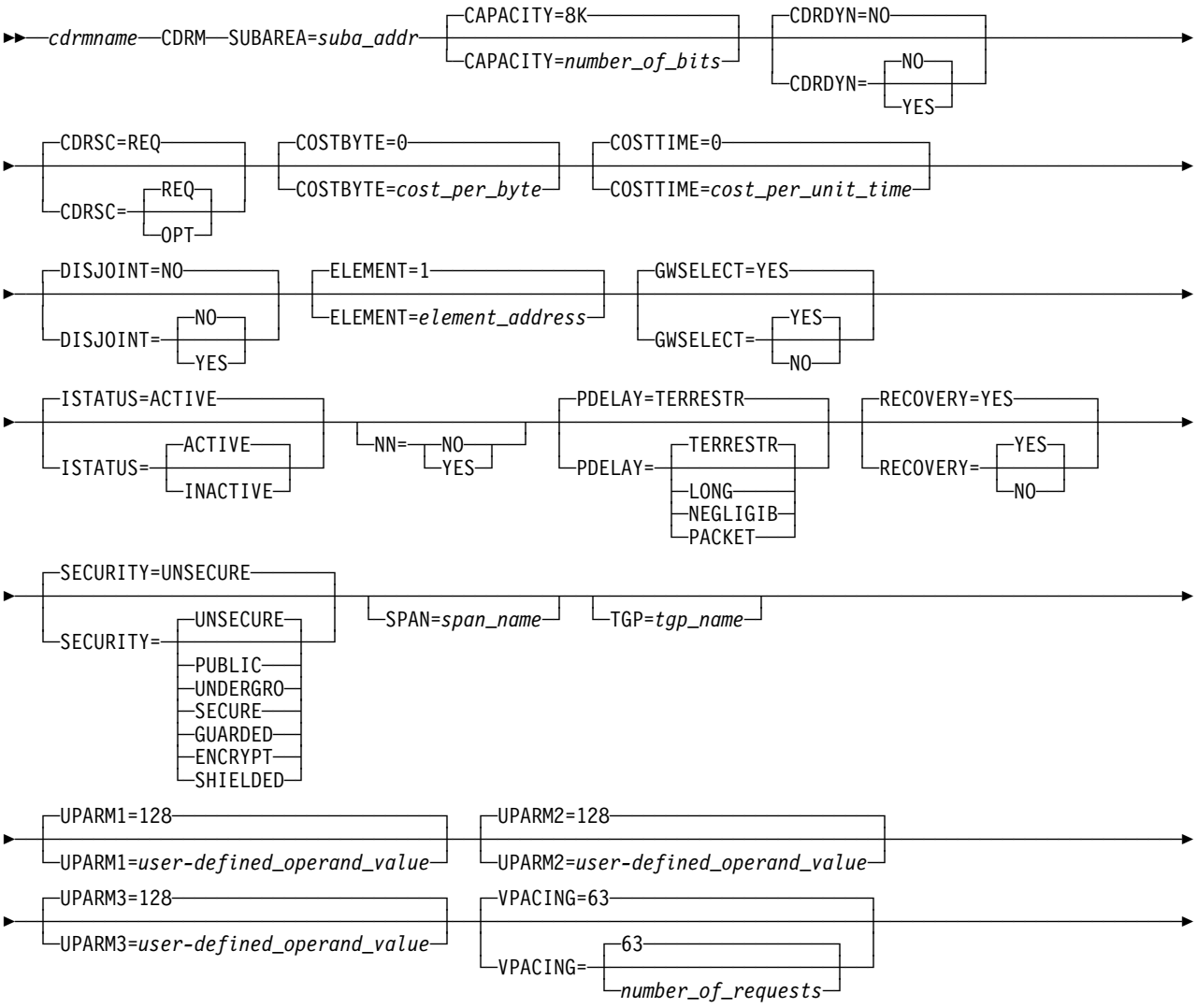
VBUILD

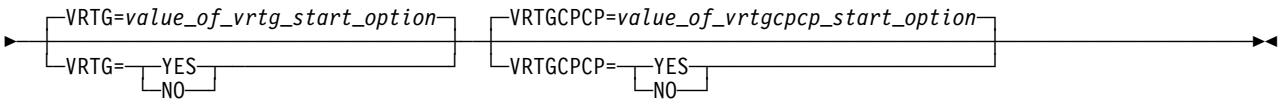


NETWORK

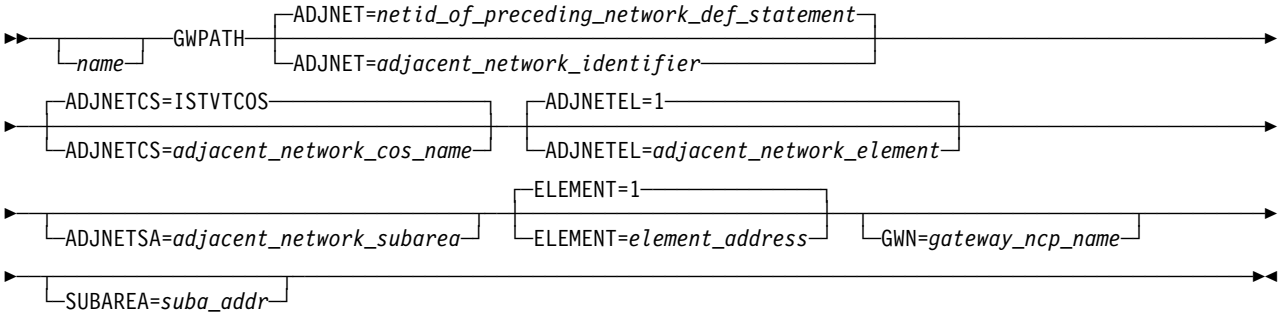


CDRM





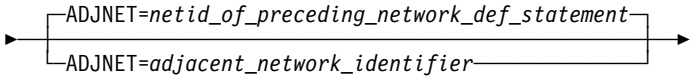
GWPATH



Operand Descriptions

The following section describes the operands you can code for the cross-domain resource manager (CDRM) major node.

ADJNET



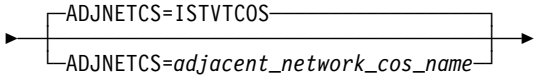
statements: GWPATH
dependency: none

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



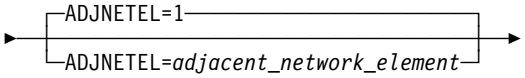
statements: GWPATH
dependency: none

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

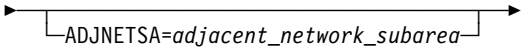


statements: GWPATH
dependency: ADJNET
range: 1–32767

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.

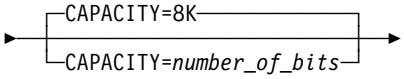
ADJNETSA



statements: GWPATH
dependency: none
range: 1–65535

specifies the subarea address of the CDRM for the adjacent network. Code ADJNETSA for all GWPATH definition statements.

CAPACITY

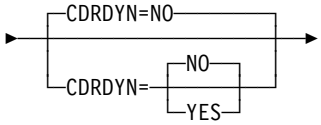


statements: CDRM
dependency: VRTG=YES coded; APPN only
range: 1Kb—1000Mb

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CDRDYN



statements: CDRM
dependency: 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

statements: CDRM
dependency: 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.

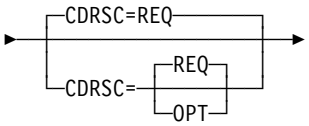
Because *cdrmname* is used in the ID operand of VTAM operator commands, it must be a name unique to this VTAM. As a result, the name fields of all CDRM definition statements that are active at one time must be unique and not conflict with other resource names in the same network. Because the CDRM definition statement in an adjacent SSCP table requires the SSCPNAME of an external CDRM, you might have to change the labels on the CDRM definition statements.

To simplify 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



statements: CDRM
dependency: 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.

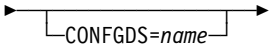
Note: CDRSC is used only for SSCP-SSCP sessions.

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.

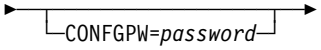
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 or a DLBL statement (VSE) that has this data definition name in the VTAM start procedure.

CONFGPW (MVS, VSE)

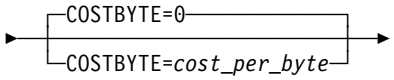


statements: VBUILD
dependency: CONFGDS

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.

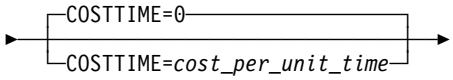
COSTBYTE



statements: CDRM
dependency: 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.

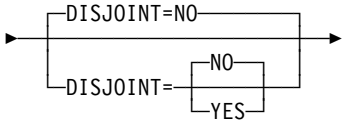
COSTTIME



statements: CDRM
dependency: 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



statements: CDRM
dependency: 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.

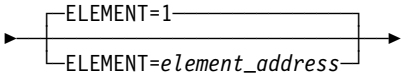
In effect, coding DISJOINT=NO restricts APPN searches, especially broadcast searches, from entering the subarea network multiple times through different interchange nodes (or even through the same interchange node.)

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



statements: CDRM, GWPATH
dependency: 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.

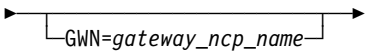
For the host CDRM, the ELEMENT value must be 1. For external CDRMs (CDRMs in other domains or networks), this number can be a decimal integer in the range 0–32767. The default value is 1 for both same- and cross-network CDRMs.

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 (MVS, VM)

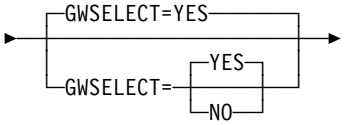


statements: GWPATH
dependency: PUNAME on BUILD statement

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 (MVS, VM)



statements: CDRM
dependency: cross-network CDRMs

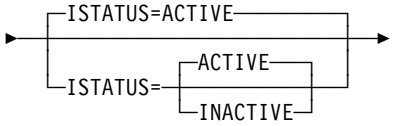
defines whether the host CDRM is to perform gateway path selection for LU-LU sessions flowing to or through this cross-network external CDRM.

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.

ISTATUS



statements: CDRM
dependency: 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:

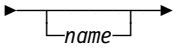
ISTATUS=ACTIVE
specifies that a session is attempted between the host in which this statement is being defined and the CDRM named by this definition statement. Any attempt to establish this session fails if either the host CDRM or the external CDRM is not active. Code ISTATUS=ACTIVE only if:

- 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



statements: VBUILD, NETWORK, GWPATH
dependency: none

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

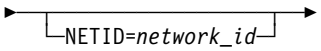
If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

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

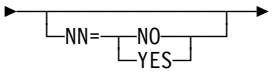


statements: NETWORK
dependency: none

specifies an optional 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.

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



statements: CDRM
dependency: 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

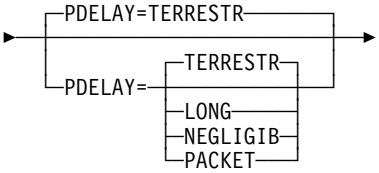
match, both the VR-based transmission group and SSCP-SSCP session will be terminated.

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



statements: CDRM
dependency: 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.

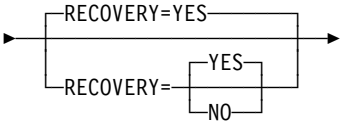
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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



statements: CDRM
dependency: none

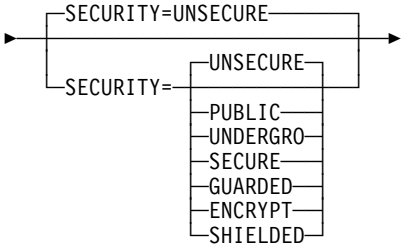
determines whether the SSCP-SSCP session between the named CDRM and the host CDRM should be restarted automatically when a session fails.

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 (MVS, VM)



statements: CDRM
dependency: 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.

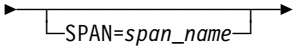
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN



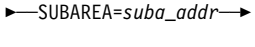
statements: CDRM
dependency: NetView

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.

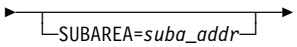
SUBAREA

On CDRM



statements:	CDRM
dependency:	none
range:	1–value of MXSUBNUM start option

On GWPATH



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:

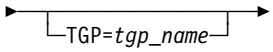
- CDRM is defined for the same network
- GWSSCP=YES in the START options and this CDRM is defined for cross-network sessions.

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

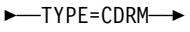


statements:	CDRM
dependency:	VRTG=YES coded; APPN only

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 495 for more information on transmission group profiles.

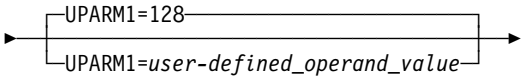
TYPE



statements: VBUILD
dependency: none

indicates a CDRM major node. This operand is required.

UPARM1

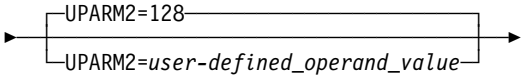


statements: CDRM
dependency: VRTG=YES coded; APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

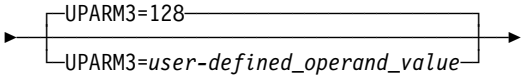
UPARM2



statements: CDRM
dependency: VRTG=YES coded; APPN only
range: 0–255

specifies a user-defined operand value.

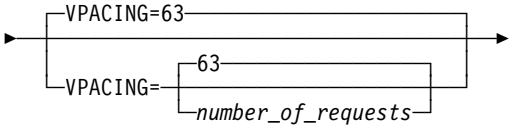
UPARM3



statements: CDRM
dependency: VRTG=YES coded; APPN only
range: 0–255

specifies a user-defined operand value.

VPACING



statements: CDRM
dependency: 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

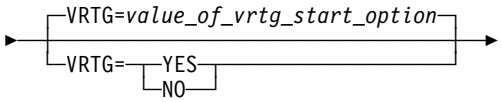
CDRM being defined is the host's own CDRM. VPACING affects only SSCP-SSCP session traffic.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

VRTG



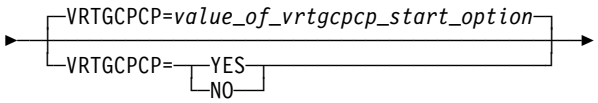
statements: CDRM
dependency: NODETYPE and HOSTSA start options coded; APPN only

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



statements: CDRM
dependency: 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

To define an external communication adapter (XCA) major node, code definition statements to define the following characteristics:

- The node type (VBUILD definition statement)
- The port used by the local area network (PORT definition statement)
- The switched nodes attached to a token-bus network, a token-ring network, a CSMA/CD 802.3 LAN, or an FDDI LAN through an IBM 3172 Interconnect Controller (GROUP, LINE, and PU definition statements).

Two types of support are defined through the external communication adapter major node:

- Subarea
- Peripheral.

The operands you need to code depend on the type of support being defined.

Refer to “External Communication Adapter (XCA) Connections” in the *VTAM Network Implementation Guide* for an example of the statements that define the external communication adapter major node.

Subarea XCA

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 166 for details on the VBUILD definition statement.

PORT Definition Statement: See “PORT Definition Statement” on page 166 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 Interconnect Controller. 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 21 (Page 1 of 2). External Communication Adapter Major Node - Subarea Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=XCA	179	R			
		CONFGDS	170				MVS,VSE
		CONFGPW	170				MVS,VSE CONFGDS
name	PORT	ADAPNO	167	R			MEDIUM
		CUADDR	171	R			
		MEDIUM	174	R			
		CAPACITY	170			8K	APPN only VNNAME VNGROUP
		COSTBYTE	170			0	APPN only VNNAME VNGROUP
		COSTTIME	171			0	APPN only VNNAME VNGROUP
		DELAY	171			0.05	
		PDELAY	175			TERRESTR	APPN only VNNAME VNGROUP
		SAPADDR	176			4	USER=SNA
		SECURITY	176			UNSECURE	APPN only VNNAME VNGROUP MVS,VM
		TGP	178				APPN only VNNAME VNGROUP
		TIMER	179			30	
		UPARM1	179			128	APPN only VNNAME VNGROUP
		UPARM2	179			128	APPN only VNNAME VNGROUP
		UPARM3	179			128	APPN only VNNAME VNGROUP
		VNGROUP	180				VNNAME; DIAL=YES
		VNNAME	181				VNGROUP; DIAL=YES
name	GROUP	DIAL	171			NO	
		SPAN	177				NetView
name	LINE	ISTATUS	173		G	ACTIVE	
		SPAN	177				NetView
		USER	180			SNA	DIAL=NO

Table 21 (Page 2 of 2). External Communication Adapter Major Node - Subarea Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PU	MACADDR	174	R			
		ISTATUS	173		G	ACTIVE	
		PUTYPE	176		G, L	5	USER=VCNS
		SAPADDR	176			4	USER=SNA
		SPAN	177				NetView
		SUBAREA	177			*in_xid_value*	
		TGN	178			1	

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.

Peripheral XCA

To define a peripheral external communication adapter node attached to a LAN through an IBM 3172 Interconnect Controller, 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 166 for details on the VBUILD definition statement.

PORT Definition Statement: See “PORT Definition Statement” on page 166 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 attached through an IBM 3172 Interconnect Controller.

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.

Table 22 (Page 1 of 3). External Communication Adapter Major Node - Peripheral Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=XCA	179	R			
		CONFGDS	170				MVS, VSE

Table 22 (Page 2 of 3). External Communication Adapter Major Node - Peripheral Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		CONFGPW	170				MVS,VSE CONFGDS
name	PORT	ADAPNO	167	R			MEDIUM
		CUADDR	171	R			
		MEDIUM	174	R			
		CAPACITY	170			8K	APPN only VNNAME VNGROUP
		COSTBYTE	170			0	APPN only VNNAME VNGROUP
		COSTTIME	171			0	APPN only VNNAME VNGROUP
		PDELAY	175			TERRESTR	APPN only VNNAME VNGROUP
		SAPADDR	176			4	
		SECURITY	176			UNSECURE	APPN only VNNAME VNGROUP
							MVS,VM
		TGP	178				APPN only VNNAME VNGROUP
		TIMER	179			30	
		UPARM1	179			128	APPN only VNNAME VNGROUP
		UPARM2	179			128	APPN only VNNAME VNGROUP
		UPARM3	179			128	APPN only VNNAME VNGROUP
		VNGROUP	180				VNNAME; DIAL=YES
		VNNAME	181				VNGROUP; DIAL=YES
name	GROUP	DIAL	171	R		NO	DIAL=YES
		AUTOGEN	168				
		DYNPU	172			NO	
		DYNPUPFX	172			CN	DYNPU=YES
		SPAN	177				NetView
name	LINE	ANSWER	168		G	ON	CALL=OUT
		CALL	169		G	IN	
		ISTATUS	173		G	ACTIVE	
		LIMRES	173		G	NO	LU 6.2
		SPAN	177				NetView
		SWNORDER	178		G	CPNAME	
name	PU	ISTATUS	173		G	ACTIVE	
		SPAN	177				NetView

Table 22 (Page 3 of 3). External Communication Adapter Major Node - Peripheral Node

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

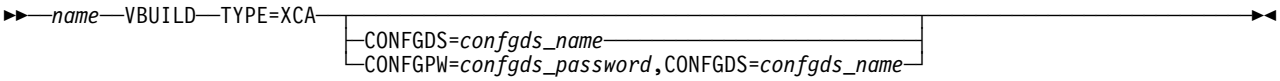
VBUILD Definition Statement: Code a VBUILD definition statement for each external communication adapter major node.

PORT Definition Statement: Code one PORT definition statement for each VBUILD TYPE=XCA definition statement. The PORT definition statement defines this host's connection to a token-bus network, a token-ring network, a CSMA/CD 802.3 LAN, or an FDDI LAN attached through an IBM 3172 Interconnect Controller. It also defines the actual adapter in the IBM 3172 Interconnect Controller through which the connection is made.

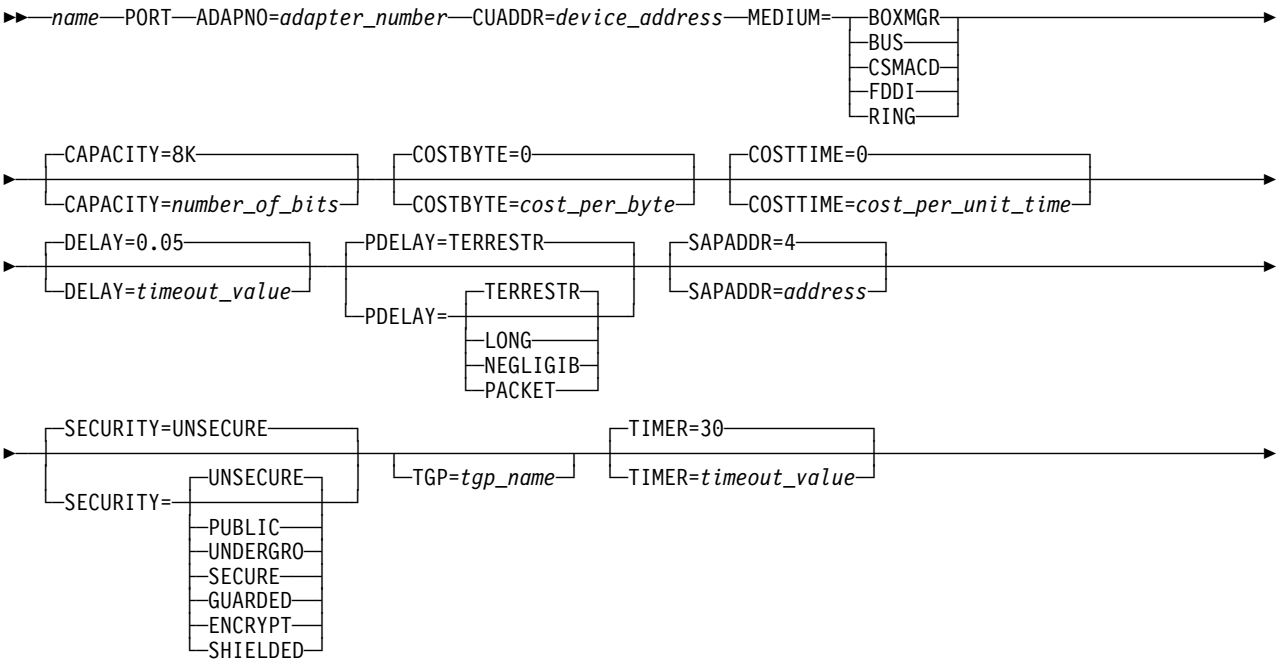
Full Syntax

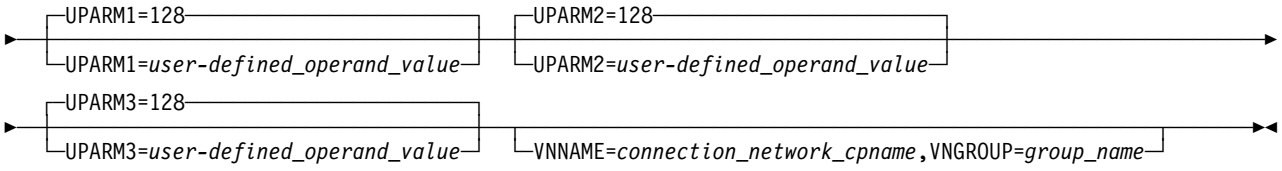
The full syntax for the external communication adapter (XCA) major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

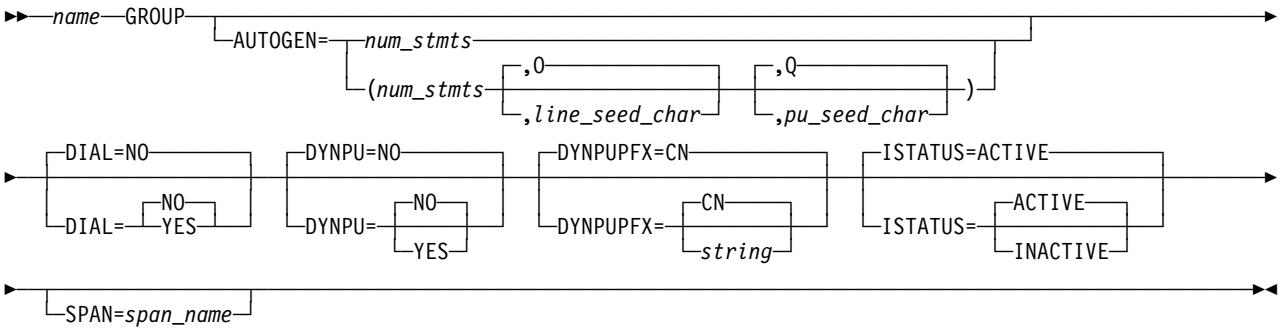


PORT

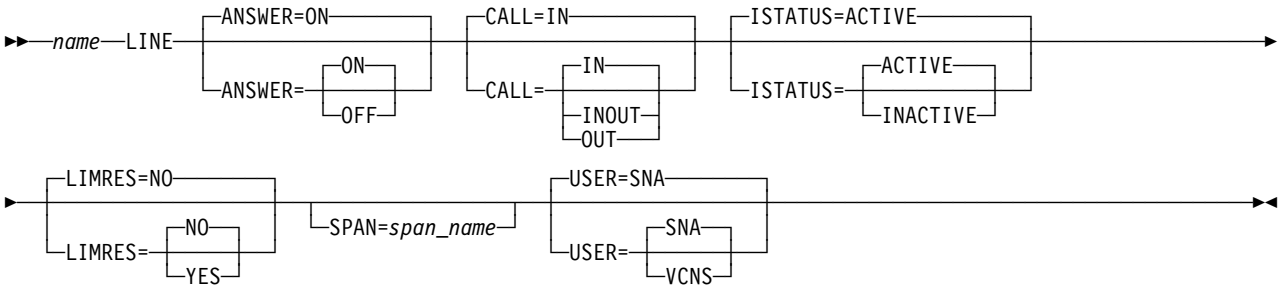




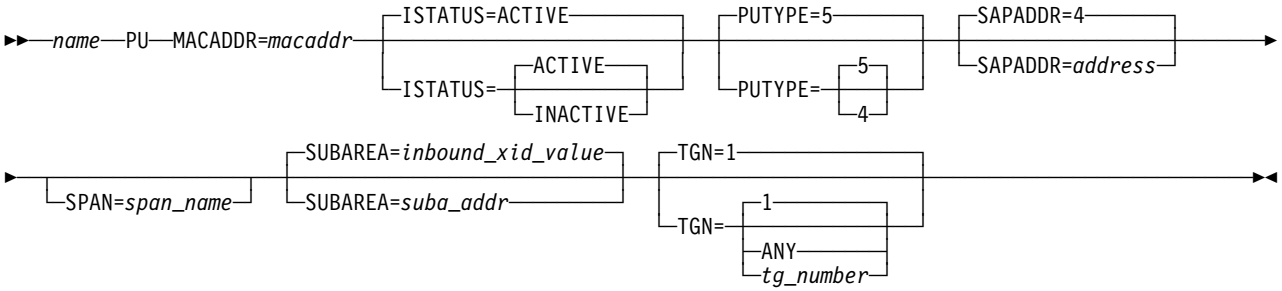
GROUP



LINE



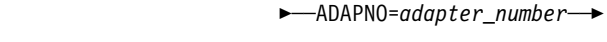
PU



Operand Descriptions

The following section describes the operands you can code for the external communication adapter (XCA) major node.

ADAPNO

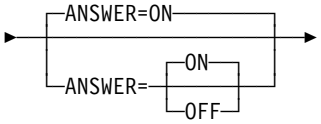


statements: PORT
dependency: MEDIUM
range: 0–127

specifies a relative adapter number assigned to each adapter by an IBM 3172 Interconnect Controller.

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 within an IBM 3172 Interconnect Controller.

ANSWER



statements: GROUP, LINE (peripheral node)
dependency: CALL=OUT

specifies whether physical units can dial in to VTAM.

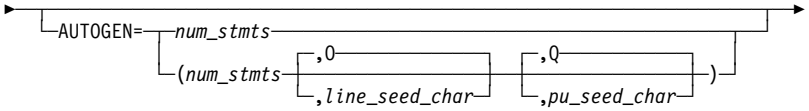
ANSWER=OFF
physical units cannot dial in to VTAM, regardless of whether the line is active or inactive.

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 "VARY ANS Command" in *VTAM 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.

AUTOGEN



statements: GROUP (peripheral node)
dependency: none

specifies that VTAM automatically generates lines and PUs.

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.

line_seed_char
defines the first character to be used by VTAM to create a name for the generated LINE statements.

pu_seed_char
defines the first character to be used by VTAM to create a name for the generated PU statements.

Note: To code *pu_seed_character* without specifying *line_seed_character*, code an extra comma before *pu_seed_character*.

VTAM generates the number of lines and PUs specified by the AUTOGEN operand. The name generation convention is *sccccnnn* where:

s Indicates the seed character defined by *line_seed_character* and *pu_seed_character*. If you do not specify a seed character, VTAM will use the defaults.

cccc
Specifies the CUA from the PORT definition statement of this major node.

nnn
Specifies a sequential number (X'0'–X'FFF').

For example, if AUTOGEN=5, no seed characters are specified, and CUA=505, VTAM generates the lines and PUs shown in Table 23.

Table 23. XCA AUTOGEN without User-Supplied Seeds

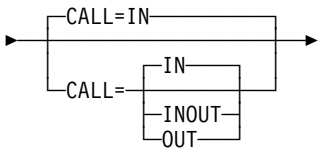
LINE	O0505000	PU	Q0505000
LINE	O0505001	PU	Q0505001
LINE	O0505002	PU	Q0505002
LINE	O0505003	PU	Q0505003
LINE	O0505004	PU	Q0505004

In another example, if AUTOGEN=(2, K, J) and CUA=505, VTAM generates the lines and PUs shown in Table 24.

Table 24. XCA AUTOGEN with User-Supplied Seeds

LINE	K0505000	PU	J0505000
LINE	K0505001	PU	J0505001

CALL



statements: GROUP, LINE (peripheral node)
dependency: none

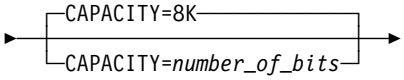
specifies whether physical units, or VTAM, or both, can initiate calls over the line represented by this LINE definition statement.

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

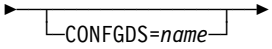


statements: PORT
dependency: VNNAME; VNGROUP; APPN only
range: 1K—1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

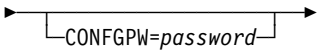
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

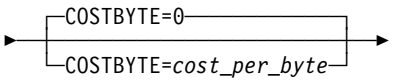


statements: VBUILD
dependency: CONFGDS

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.

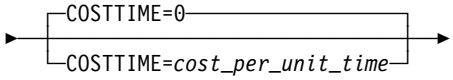
COSTBYTE



statements: PORT
dependency: VNNAME; VNGROUP; 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.

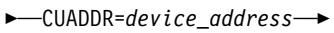
COSTTIME



statements: PORT
dependency: VNNAME; VNGROUP; 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.

CUADDR



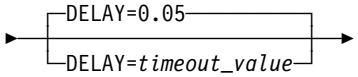
statements: PORT
dependency: none
range: X'000'–X'FFFF'

defines a 3-digit or 4-digit (MVS) hexadecimal channel unit address (CUA) used for the attached LAN. The channel unit address should be within the address range defined to the operating system. The address should also be a unique address.

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.

DELAY

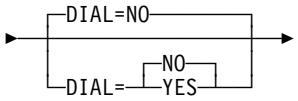


statements: PORT
dependency: none
range: 0–9.999

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, set this value to 0. Refer to “DELAY Operand” in the *VTAM Network Implementation Guide* for more information.

DIAL



statements: GROUP
dependency: none

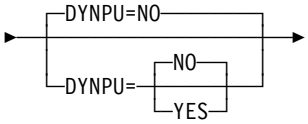
specifies whether the lines under the GROUP statement require switched line control protocols.

Note: You must code DIAL=YES for peripheral nodes.

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.

DYNPU



statements: GROUP (peripheral node)
dependency: 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.

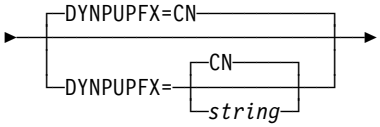
A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:

- 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



statements: GROUP (peripheral node)
dependency: 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.

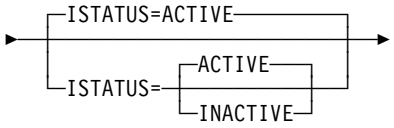
Note: This operand is valid only if DYNPU is specified as YES.

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 "Format of Definition Statements" on page 10 for naming conventions.

ISTATUS



statements: GROUP, LINE, PU
dependency: none

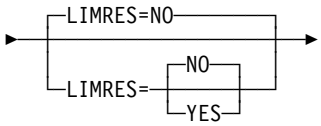
specifies whether a line or PU is to be activated after the first activation of the external communication adapter major node.

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 "VARY ACT Command" in *VTAM Operation* for information on using the VARY ACT command.

LIMRES



statements: GROUP, LINE (peripheral node)
dependency: LU 6.2 only

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 can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

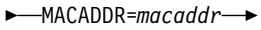
If you do not code a value for LIMQSINT, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

This operand applies to LU 6.2 conversations only.

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.

MACADDR

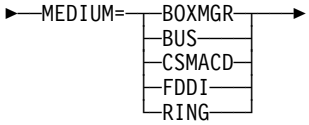


statements: PU (subarea node)
dependency: subarea XCA only

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. X'000000000000' is not a valid MACADDR value.

MEDIUM



statements: PORT
dependency: none

identifies the type of LAN being defined. This operand is required. The valid values for this operand are BUS, RING, CSMACD, FDDI, and BOXMGR.

MEDIUM=BOXMGR
defines a box manager. A box manager allows an IBM 3172 Interconnect Controller to have minimal network management. It also allows generic alerts to flow from an IBM 3172 Interconnect Controller to the NetView program.

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 “External Communication Adapter (XCA) Connections” in the *VTAM Network Implementation Guide* for an example of coding a box manager definition.

MEDIUM=BUS
defines a connection to a token-bus network through an IBM 3172 Interconnect Controller.

MEDIUM=CSMACD
defines a connection to a CSMA/CD 802.3 LAN through an IBM 3172 Interconnect Controller.

MEDIUM=FDDI
defines a connection to an FDDI LAN through an IBM 3172 Interconnect Controller.

MEDIUM=RING
defines a connection to a token-ring network through an IBM 3172 Interconnect Controller.

name

statements: VBUILD, PORT, GROUP, LINE, PU
dependency: none

On VBUILD: *name* provides the required and unique name for this major node.

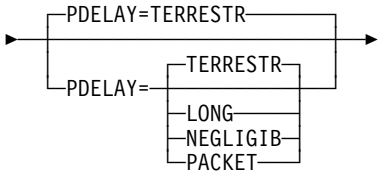
On PORT: *name* provides the required name associated with a particular adapter in an IBM 3172 Interconnect Controller.

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.

PDELAY



statements: PORT
dependency: VNNAME; VNGROUP; 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.

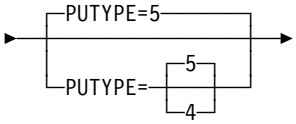
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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



statements: GROUP, LINE, PU (subarea node)
dependency: USER=VCNS

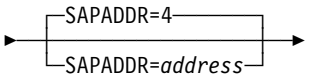
specifies the physical unit type.

Note: If USER=VCNS is specified on the LINE definition statement, the PUTYPE operand is not valid.

PUTYPE=4
specifies a connection to a 37x5 communication controller running NCP/Token-Ring interconnection (NTRI). Specify PUTYPE=4 only if MEDIUM=RING on the PORT definition statement.

PUTYPE=5
specifies a connection to a host through an IBM 3172 Interconnect Controller.

SAPADDR



statements: PORT, PU (subarea node)
dependency: USER=SNA (LINE statement)
range: 4–252

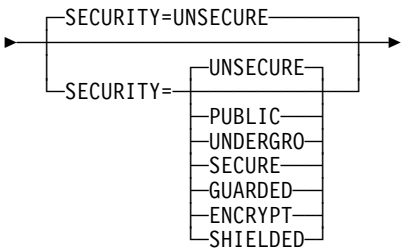
On PORT: specifies the service access point (SAP) address for the connection to a LAN attached through the IBM 3172 Interconnect Controller that this major node defines.

The value you specify must be a multiple of 4. 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 connection to the LAN attached through an IBM 3172 Interconnect Controller that this PU definition statement defines.

SECURITY (MVS, VM)



statements: PORT
dependency: VNNAME; VNGROUP; 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.

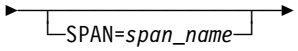
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN

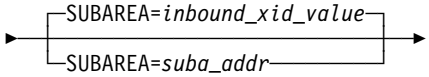


statements: GROUP, LINE, PU
dependency: NetView

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.

SUBAREA

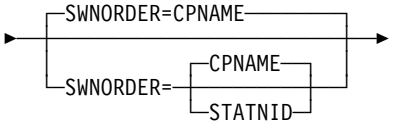


statements: PU (subarea node)
dependency: none
range: 1–value of the MXSUBNUM start option

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



statements: GROUP, LINE
dependency: none

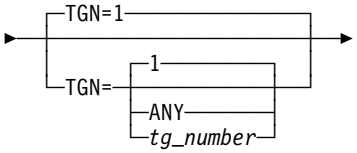
specifies the way VTAM locates a switched PU. If you do not code SWNORDER, VTAM searches for a PU by the CPNAME first. If VTAM does not find the PU by the CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

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



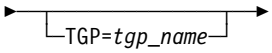
statements: PU (subarea node)
dependency: none
range: 1–255

specifies the transmission group (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 transmission group 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.

TGP

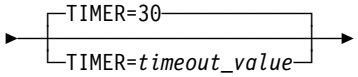


statements: PORT
dependency: VNNAME; VNGROUP; APPN only

specifies the name of a transmission group (TG) profile definition used for the connection network. The characteristic defined in the TG profile (along with any modifiers specified on this statement) become the characteristics of the connection network. If a *tg_name* is not specified or has not been activated, the default TG profile operand values are used.

See “APPN Transmission Group Profile” on page 495 for more information on transmission group profiles.

TIMER

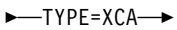


statements: PORT
dependency: none
range: 1–255

specifies the number of seconds that VTAM will wait for a response from an IBM 3172 Interconnect Controller 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 for the 3172 controller: the calculation of the inactivity timer multiplied by the number of retransmissions for the 3172 controller should be less than the value coded for TIMER.

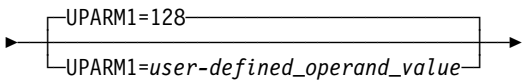
TYPE



statements: VBUILD
dependency: none

indicates an external communication adapter major node. This operand is required.

UPARM1

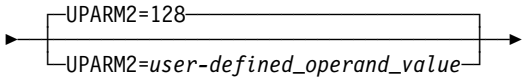


statements: PORT
dependency: VNNAME; VNGROUP; APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

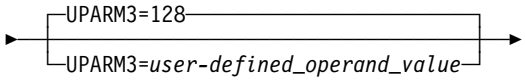
UPARM2



statements: PORT
dependency: VNNAME; VNGROUP; APPN only
range: 0–255

specifies a user-defined operand value.

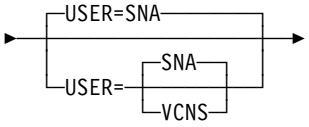
UPARM3



statements: PORT
dependency: VNNAME; VNGROUP; APPN only
range: 0–255

specifies a user-defined operand value.

USER



statements: LINE (subarea node)
dependency: 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 an IBM 3172 Interconnect Controller for use by the LAN.

Note: The MACADDR and SAPADDR operands on the accompanying PU definition statement provide the remote PU's address.

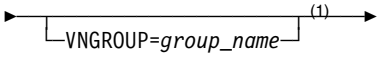
USER=VCNS
specifies that VCNS uses the line to access an IBM 3172 Interconnect Controller for use by the LAN.

When USER=VCNS, code only one LINE definition statement for each external communication adapter 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.

VNGROUP



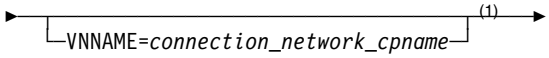
Note:
[1] Code VNGROUP if you code VNNAME.

statements: PORT
dependency: 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



Note:
[1] Code VNNAME if you code VNGROUP.

statements: PORT
dependency: 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. 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.

LAN Major Node (VM, VSE)

A local area network (LAN) major node defines a LAN attached through an integrated communication adapter (ICA).

To define a LAN major node, code definition statements to specify the following characteristics:

- The node type (VBUILD definition statement)
- The port the LAN uses (PORT definition statement)
- The subarea and peripheral nodes, including processors attached to the LAN (GROUP, LINE, PU definition statements).

Refer to the *VTAM Network Implementation Guide* for examples of the definition statements that define local area network major nodes.

The following terms are used in defining a LAN major node:

LAN An IBM Token-Ring local area network

IBM Token-Ring Subsystem Controller
A pair of input/output processor (IOP) and input/output adapter (IOA) cards that provide a connection to a LAN.

LAN Major Node for Subarea Nodes

A processor with an IBM Token-Ring Subsystem Controller, a processor with a multi-protocol communication subsystem or a 37x5 running NCP/Token-Ring interconnection attached to the token-ring is defined to VTAM by the LINE and PU definition statements. The LINE and PU definition statements must be specified as pairs in a nonswitched line group with a GROUP statement.

To define a LAN major node for subarea nodes, code the following definition statements:

VBUILD
PORT
GROUP
LINE
PU

VBUILD Definition Statement: See “VBUILD Definition Statement” on page 185 for details on the VBUILD definition statement.

PORT Definition Statement: See “PORT Definition Statement” on page 185 for details on the PORT definition statement.

GROUP Definition Statement: Code a GROUP definition statement to define a line group for processors or 37x5s attached to the token-ring.

The major node can contain different line groups, and each line group must have a unique GROUP definition statement and contain a unique set of minor node statements.

LINE Definition Statement: Code a LINE definition statement for each processor or 37x5 on the token-ring.

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 processor or 37x5 with which VTAM will communicate over this token-ring.

Table 25. LAN Major Node Definition Statements and Operands - Subarea

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LAN	201	R			
		CONFGDS	190				VSE
		CONFGPW	190				VSE CONFGDS
name	PORT	CUADDR	191	R			
		BUFFERS	188			(5,5)	
		LANCON	193			(,5)	
		MACADDR	196				
		MAXDATA	196			0	
		MAXSTN	197				
		SAPADDR	199			4	
name	GROUP	DIAL=NO	191	R			nonswitched lines
		LNCTL=SDLC	196				
		SPAN	200				NetView
name	LINE	ISTATUS	193		G	ACTIVE	
		SPAN	200				NetView
name	PU	MACADDR	196	R			
		ISTATUS	193		G	ACTIVE	
		LANACK	193		G,L	(0,0)	
		LANCON	193		G,L		
		LANINACT	194		G,L	4.8	
		LANRESP	194		G,L		
		LANSDWDW	195		G,L	(2,1)	
		PUTYPE	198		G,L	5	
		SAPADDR	199			4	
		SPAN	200				NetView
		SUBAREA	200			*in_xid_value*	PU types 4, 5
		TGN	200			1	

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.

LAN Major Node for Peripheral Nodes

A peripheral node attached to the token-ring is defined to VTAM by the LINE and PU definition statements. The LINE and PU definition statements must be specified in pairs in a switched line group with a GROUP definition statement.

To define a LAN major node for peripheral nodes, code the following definition statements:

VBUILD
PORT
GROUP
LINE
PU

VBUILD Definition Statement: See "VBUILD Definition Statement" on page 185 for details on the VBUILD definition statement.

PORT Definition Statement: See "PORT Definition Statement" on page 185 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 attached to the token-ring.

The major node can contain different line groups, and each line group must have a unique GROUP definition statement and contain a unique set of minor node 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 line.

Table 26 (Page 1 of 2). LAN Major Node Definition Statements and Operands - Peripheral

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LAN	201	R			
		CONFGDS	190				VSE
		CONFGPW	190				VSE CONFGDS
name	PORT	CUADDR	191	R			
		BUFFERS	188			(5,5)	
		CAPACITY	190			8K	APPN only
		COSTBYTE	190			0	APPN only
		COSTTIME	191			0	APPN only
		LANCON	193			(,5)	
		MACADDR	196				
		MAXDATA	196			0	
		MAXSTN	197				
		PDELAY	198			TERRESTR	APPN only
		SAPADDR	199			4	
		SECURITY	199			UNSECURE	APPN only; VM
		TGP	200				APPN only

Table 26 (Page 2 of 2). LAN Major Node Definition Statements and Operands - Peripheral

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		UPARM1	201			128	APPN only
		UPARM2	201			128	APPN only
		UPARM3	201			128	APPN only
		VNGROUP	201				VNNAME; DIAL=YES
		VNNAME	202				VNGROUP; DIAL=YES
name	GROUP	DIAL=YES	191	R			switched lines
		AUTOGEN	187				
		DYNPU	192			NO	
		DYNPUPFX	192			CN	DYNPU=YES
		LNCTL=SDLC	196				
		SPAN	200				NetView
name	LINE	ANSWER	187		G	ON	CALL=OUT
		CALL	189		G	IN	
		ISTATUS	193		G	ACTIVE	
		LIMRES	195		G	NO	LU 6.2 only
		SPAN	200				NetView
name	PU	ISTATUS	193	G		ACTIVE	
		SPAN	200				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.

Coding Notes

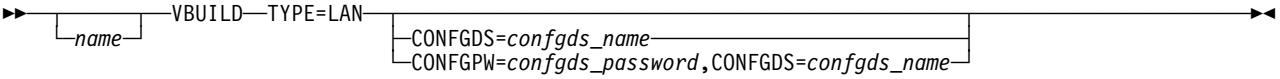
VBUILD Definition Statement: Code one VBUILD definition statement for each local area network major node.

PORT Definition Statement: Code one PORT definition statement to identify the connection between VTAM and a LAN. The PORT definition statement identifies the address of the access protocol group in the IBM Token-Ring Subsystem Controller or the multi-protocol communication subsystem of the processor.

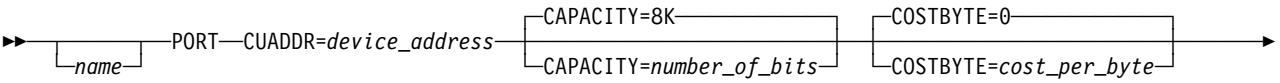
Full Syntax

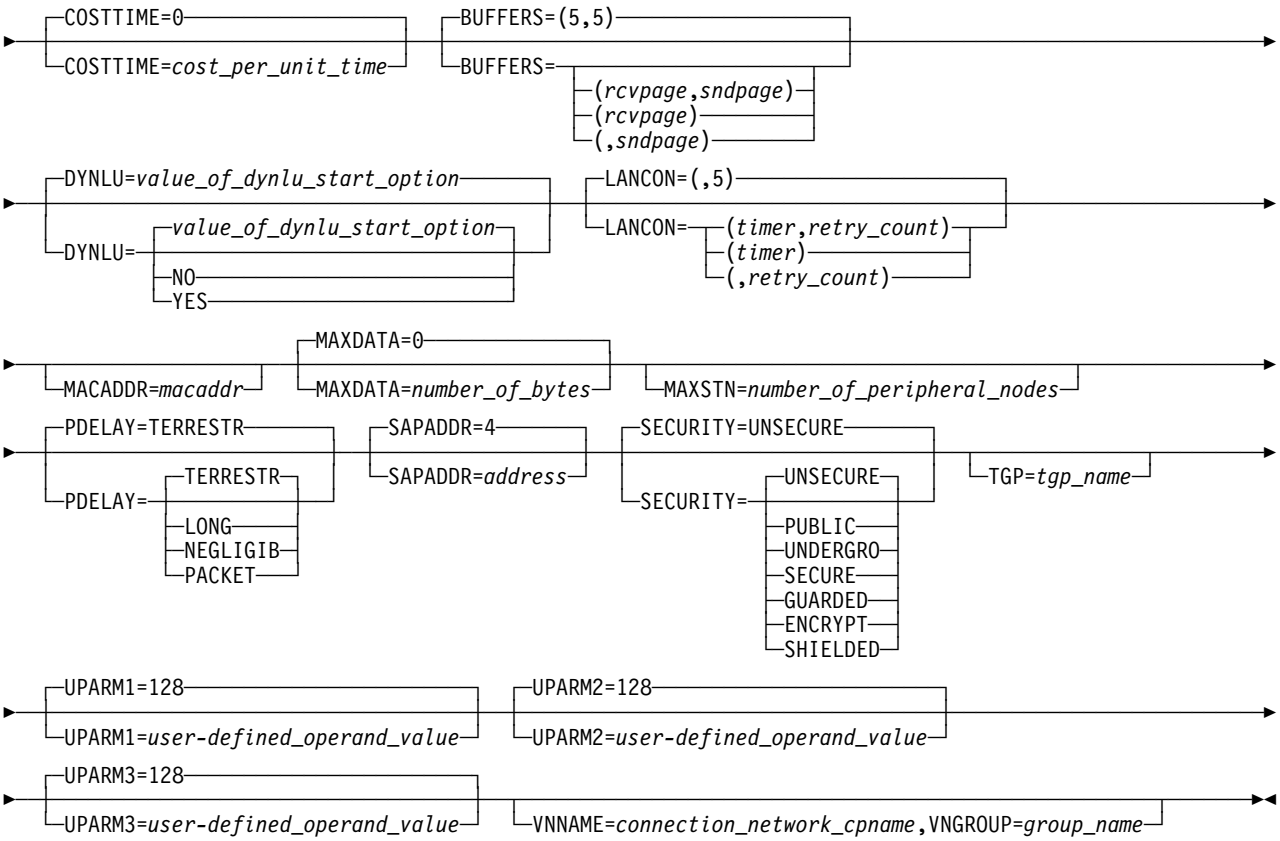
The full syntax for the packet major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

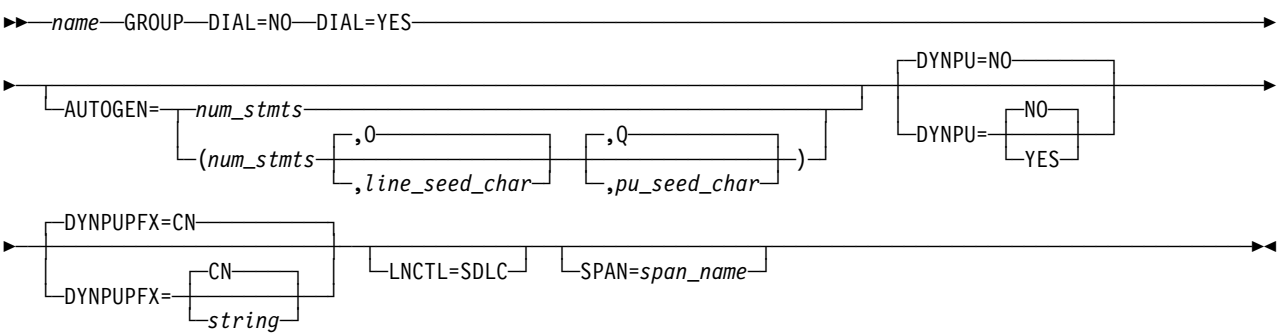


PORT

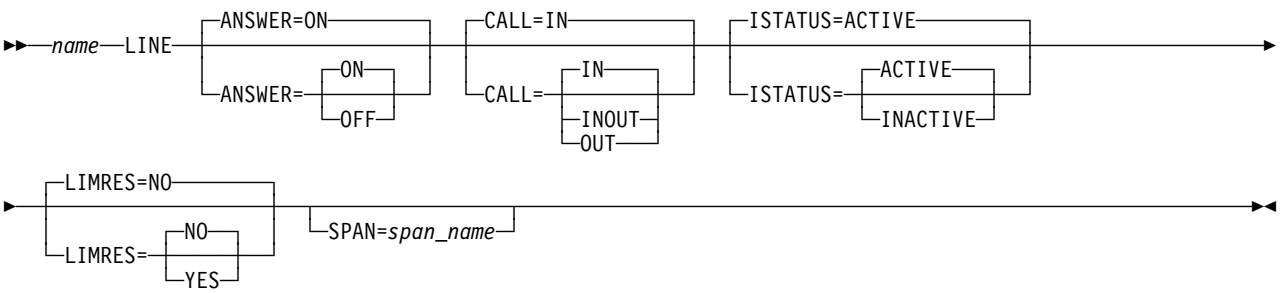




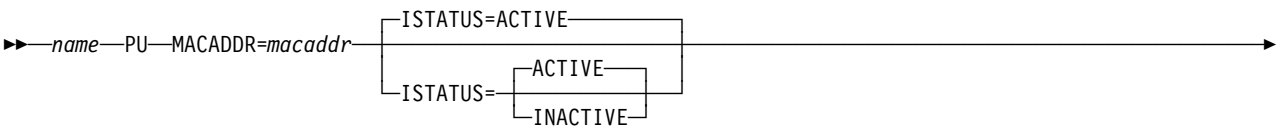
GROUP

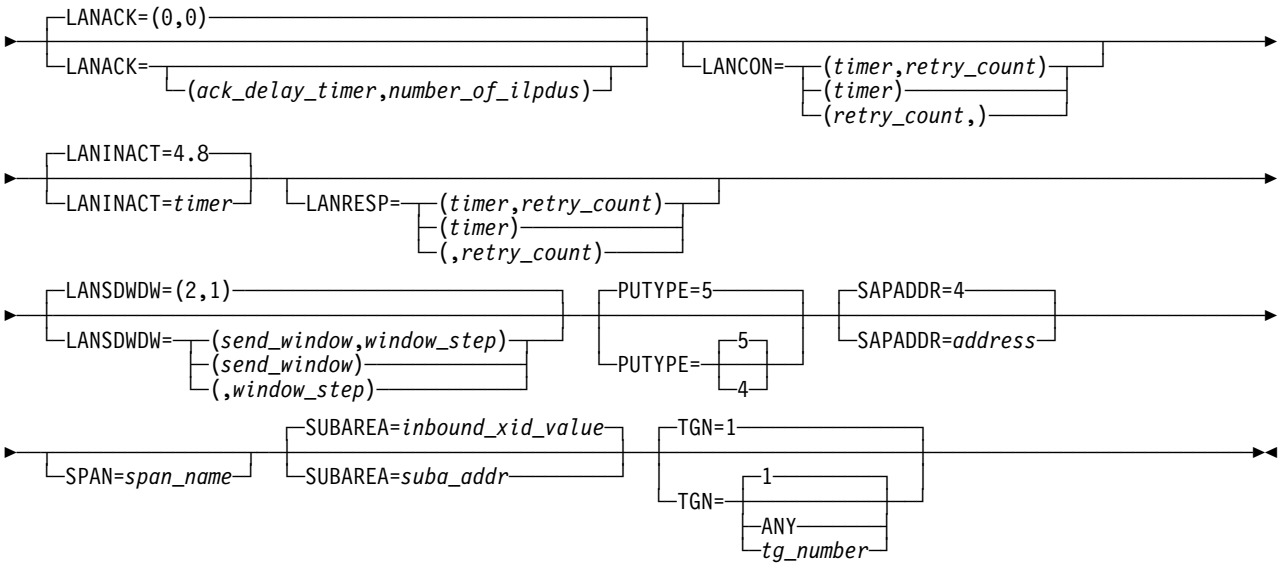


LINE



PU

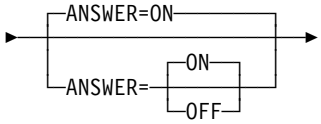




Operand Descriptions

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

ANSWER



statements: GROUP, LINE
dependency: CALL=OUT

specifies whether physical units can dial in to VTAM.

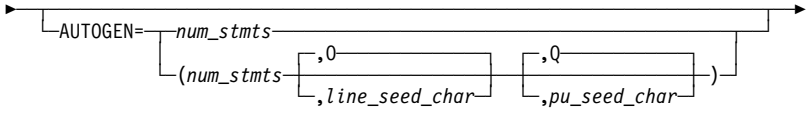
ANSWER=OFF
specifies that physical units cannot dial in to VTAM, regardless of whether the line is active or inactive.

ANSWER=ON
specifies that 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. See *VTAM Operation* for information on using the VARY ANS command.

Note: When CALL=OUT, the ANSWER operand is ignored, and the operand receives a message that this operand is not valid.

AUTOGEN



statements: GROUP
dependency: none

specifies that VTAM automatically generates lines and PUs.

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.

line_seed_char
defines the first character to be used by VTAM to create a name for the generated LINE statements.

pu_seed_char
defines the first character to be used by VTAM to create a name for the generated PU statements.

Note: To code *pu_seed_character* without specifying *line_seed_character*, code an extra comma before *pu_seed_character*.

VTAM generates the number of lines and PUs specified by the AUTOGEN operand. The name generation convention is *sccccnnn* where:

s Indicates a seed character which you can specify as an AUTOGEN parameter. If you do not specify a value for *s*, VTAM will assign one. The default seeds are *O* and *Q* as described above.

cccc
Specifies the CUA from the PORT definition statement of this major node

nnn
Specifies a sequential number (0-FFF)

For example, if CUA=**505**, VTAM generates the lines and PUs shown in Table 27 if AUTOGEN=**5** and you do not specify any seeds.

Table 27. LAN AUTOGEN Without User-Supplied Seeds

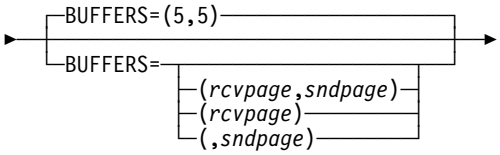
LINE	O0505000	PU	P0505000
LINE	O0505001	PU	P0505001
LINE	O0505002	PU	P0505002
LINE	O0505003	PU	P0505003
LINE	O0505004	PU	P0505004

In another example, when you specify AUTOGEN=(2, K, J), and if CUA=**505**, VTAM generates the lines and PUs shown in Table 28.

Table 28. LAN AUTOGEN With User-Supplied Seeds

LINE	K0505000	PU	J0505000
LINE	K0505001	PU	J0505001

BUFFERS



statements: PORT
dependency: none
range: 3–64 pages

defines the amount of storage preallocated to inbound and outbound data transfer with the IBM Token-Ring Subsystem Controller.

The valid range for *rcvpage* and *sndpage* is 3–64 pages. The suggested minimum number of pages is the greater of 3 or the value calculated using the following formula:

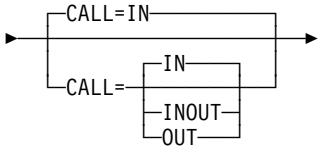
```
((2 times maximum frame size)/4K ) + 1
```

rcvpage|5
specifies the number of 4K pages that are allocated for receiving data from the IBM Token-Ring Subsystem Controller. The resulting inbound capacity is *rcvpage* × 4K. Specify a number large enough that the resulting capacity can contain at least two maximum logical link control (LLC) protocol data units (LPDUs) to be received from the IBM Token-Ring Subsystem Controller.

sndpage|5
specifies the number of 4K pages that are allocated for sending data to the IBM Token-Ring Subsystem Controller. The resulting outbound capacity is *sndpage* times 4K. Specify a number large enough that the resulting capacity can contain at least two maximum LPDUs being sent to the IBM Token-Ring Subsystem Controller.

Note: The inbound and outbound buffers are allocated in separate storage. Therefore, you can specify different values for *rcvpage* and *sndpage*.

CALL



statements: GROUP, LINE
dependency: none

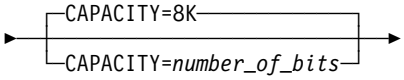
specifies whether physical units, or VTAM, or both, can initiate calls over the line represented by this LINE definition statement. The CALL operand acts as a security device.

CALL=IN
specifies that the line is to be used only for incoming calls (the physical units 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 calls the physical unit).

CAPACITY

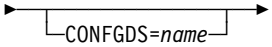


statements: PORT
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CONFGDS (VSE)

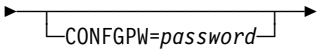


statements: VBUILD
dependency: none

is a 1–8 character file name that identifies the configuration restart file defined by the user for this major node.

Include in the VTAM start procedure a DLBL statement that has this data definition name.

CONFGPW (VSE)

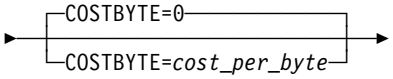


statements: VBUILD
dependency: none

specifies the 1–7 character alphanumeric password, if required, for VTAM to gain access to the configuration restart file. If CONFGPW is not specified, but is required by VSAM, VSAM prompts for the correct password when VTAM attempts to open the file.

You can specify this operand only if the CONFGDS operand is specified.

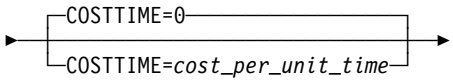
COSTBYTE



statements: PORT
dependency: 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.

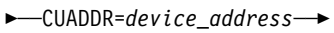
COSTTIME



statements: PORT
dependency: 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.

CUADDR



statements: PORT
dependency: none
range: first digit: X'0–F'
second, third digits: X'00–FB'

defines the required four 3-digit hexadecimal channel unit addresses used for the attached LAN. VTAM uses *cua* and the next three sequential addresses as channel unit addresses. For example, if you specify CUADDR=848, VTAM assigns 848, 849, 84A, and 84B as the channel unit addresses.

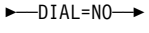
The first digit of the channel unit address is the channel number and it has a range X'0'–X'F'. The second and third digits are the subchannel address, in the range X'00'–X'FB'. For example, a channel address of X'012' is coded as CUADDR=012.

If there is more than one connection (access protocol group) between VTAM and the subsystem, the values coded for the CUADDR operand must be different for each VTAM connected to the subsystem. See the *VTAM Network Implementation Guide* for more information. When VTAM attaches to the LAN, it verifies that the device at each of the four channel unit addresses is defined to the operating system as follows:

VM As a 3088 device
VSE As a CETI device.

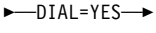
DIAL

For Subarea nodes:



statements: GROUP
dependency: nonswitched lines

For Peripheral nodes:



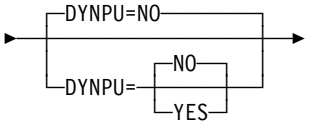
statements: GROUP
dependency: switched lines

specifies whether or not the lines in the group require switched line control protocols.

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.

DYNPU



statements: GROUP (switched)
dependency: 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.

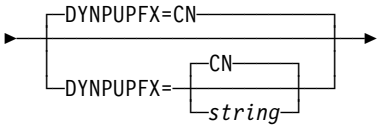
A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:

- 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



statements: GROUP (switched)
dependency: 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.

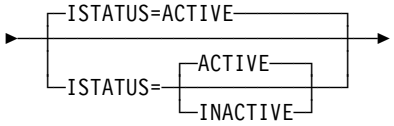
Note: This operand is valid only if DYNPU is specified as YES.

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 “Format of Definition Statements” on page 10 for naming conventions.

ISTATUS



statements: GROUP, LINE, PU
dependency: none

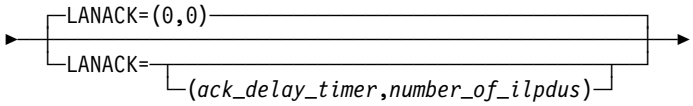
specifies whether the resource is to be activated after the first activation of the LAN major node.

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 "VARY ACT Command" in *VTAM Operation* for information on using the VARY ACT command.

LANACK



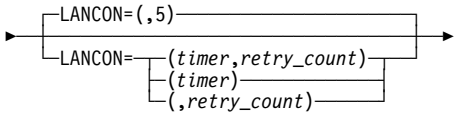
statements: GROUP, LINE, PU
dependency: none
range: *ack_delay_timer*: 0–25.5 seconds
number_of_ilpdus: 0–127 seconds

specifies the acknowledgement delay timer (*ack_delay_timer*) and number of information-LPDUs (I-LPDUs) (*number_of_ilpdus*) received prior to sending an acknowledgement. If either the timer expires or the count is reached, an acknowledgement is sent for any outstanding data.

A value of 0 for *ack_delay_timer* means that the timer is disabled. Note that you cannot specify 0 or take the default for just the timer or the counter. If you code one as 0, you must code 0 for both. For example, you cannot code LANACK=(0,2). If you want the timer to be 0, you must code LANACK=(0,0). Similarly, you cannot code LANACK=(3,0). If you want the counter to be 0, you must code LANACK=(0,0).

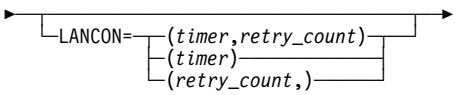
LANCON

For PORT:



statements: PORT
dependency: none
range: *timer*: 0–25.5
retry_count: 0–255

For GROUP, LINE, PU:



statements: GROUP, LINE, PU
dependency: none
range: *timer*: 0–25.5
retry_count: 0–255

specifies the timer (*timer*) and the retry count (*retry_count*) for the connection LPDUs.

timer
specifies the time, in seconds or tenths of seconds, that elapses before a retry is attempted during connection or disconnection of a link station. If you specify 0 or do not specify a value for *timer*, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

timer should have a value sufficiently large enough to account for the propagational delay of the network. The LPDU response must be received before the timer expires. A value too small results in the reporting of errors that are not valid.

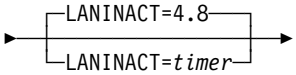
retry_count
specifies the number of times to retry a transmission during connection or disconnection of a link station. A value of 0 means that no retry will be attempted.

retry_count must be greater than 1 when the network configuration incorporates the use of bridges. A value of 1 or less prevents the “all routes” broadcast test frame from being sent.

When the number of retries has been reached and the timer has elapsed for each retry, the link station is considered to be inoperative.

If you do not code LANCON or LANRESP operands on the PU definition statement, the *timer* and *retry_count* values you specify on the PORT definition statement are used as the default values.

LANINACT

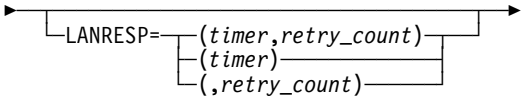


statements: GROUP, LINE, PU
range: 0–25.5 seconds

specifies the timer inactivity in seconds or tenths of seconds. *timer* specifies the time used to determine an inactive condition on a link station.

If you specify 0, the timer is disabled.

LANRESP



statements: GROUP, LINE, PU
dependency: none
range: *timer*: 0–25.5
retry_count: 0–255

specifies the timer (*timer*) and the retry count (*retry_count*) for a station in connected state.

timer
specifies the time, in seconds or tenths of seconds, that elapses before a retry is attempted when a station is connected.

If you specify 0, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

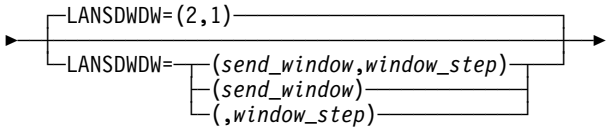
retry_count
specifies the number of times to retry a transmission while a station is connected.

A value of 0 means that no retry will be attempted.

When the number of retries has been reached and the timer has elapsed for each retry, the link station is considered to be inoperative.

If you do not specify LANRESP for this physical unit, the default values for the timer and the retry count are the values that you specify on the LANCON operand of the PORT definition statement.

LANSDWDW

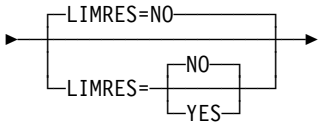


statements: GROUP, LINE, PU
dependency: none
range: *send_window*: 1–127
window_step: 0–the value of *send_window*

specifies the send window and window step. The send window specifies the maximum number of sequentially numbered I-LPDUs that the link station can have outstanding at any time. The window step specifies the number of sequentially numbered I-LPDUs that a link station must receive before increasing the local window.

A value of 0 for the window step specifies that the send window is static and dynamic windowing is disabled.

LIMRES



statements: GROUP, LINE
dependency: applies to LU 6.2 only

specifies whether a line or a 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 can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

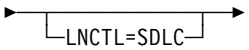
If you do not specify a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions remain active even after all conversations have ceased.

This operand applies to LU 6.2 conversations only.

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.

LNCTL=SDLC

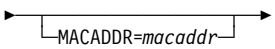


statements: GROUP
dependency: none

specifies that the lines under a LAN port appear to VTAM as SDLC lines.

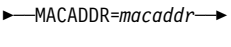
MACADDR

For PORT:



statements: PORT
dependency: none

For PU:



statements: PU
dependency: none

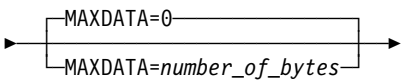
On PORT: specifies the 12-digit hexadecimal medium access control (MAC) address for the connection to the LAN that this major node defines. If you do not specify MACADDR on the PORT definition statement, the preset MAC address of the IBM Token-Ring Subsystem Controller is used.

When information flows from the LAN to VTAM, the MACADDR operand identifies the subsystem to which the information should be routed. If there are multiple subsystems connected to the same LAN, the values of the MACADDR operand must be different for each subsystem.

On PU: specifies the 12-digit hexadecimal medium access control (MAC) address for the 37x5 communication controller or the processor with an IBM Token-Ring Subsystem Controller or the multi-protocol communication subsystem on the ring that this PU definition statement defines.

On the PU or PORT definition statement, if you specify the MAC address as an 8-digit hexadecimal number, VTAM prefixes the number with 4000 to form the 12-digit address. X'000000000000' is not a valid MACADDR value.

MAXDATA



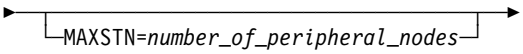
statements: PORT
dependency: none
range: 0–65535

specifies the maximum number of bytes in the information field of an LPDU that can be transmitted on the LAN.

If you code MAXDATA=0 or use it by default, VTAM requests the largest frame size supported by the IBM Token-Ring Subsystem Controller adapter.

Note: If you specify a non-zero value for MAXDATA, it must be within the minimum and maximum value range allowed by the IBM Token-Ring Subsystem Controller or the multi-protocol communication subsystem. Consult the appropriate documentation for your subsystem to determine the allowed range of values.

MAXSTN



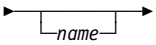
statements: GROUP
dependency: none
range: 1–65535

specifies the maximum number of peripheral nodes that can be connected on the LAN.

The default value for MAXSTN is the number of peripheral nodes that are defined in the LAN major node.

Note: The value specified for MAXSTN cannot exceed the maximum value allowed by the IBM Token-Ring Subsystem Controller or the multi-protocol communication subsystem. Consult the appropriate documentation for your subsystem to determine the maximum value that can be coded.

name



statements: VBUILD, PORT

statements: GROUP, LINE, PU

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.

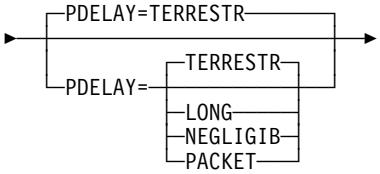
On PORT: provides the name associated with a particular adapter on a processor with an IBM Token-Ring Subsystem Controller or a processor with a multi-protocol communication subsystem. Although a name is optional, it is recommended that you code one because the node can then be identified by name when messages are issued regarding the node's status.

On GROUP: provides the required minor node name of the line group represented by this definition statement.

On LINE: provides the required minor node name of the line represented by this definition statement.

On PU: provides the required minor node name of the physical unit represented by this definition statement.

PDELAY



statements: PORT
dependency: 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.

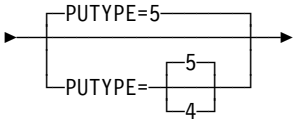
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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



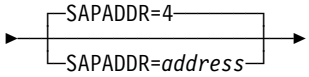
statements: GROUP, LINE, PU
dependency: none

specifies the physical unit type.

PUTYPE=4
specifies a connection to a 37x5 communication controller running NCP/Token-Ring interconnection.

PUTYPE=5
specifies that a processor with an IBM Token-Ring Subsystem Controller or a multi-protocol communication subsystem is a type 5 node.

SAPADDR



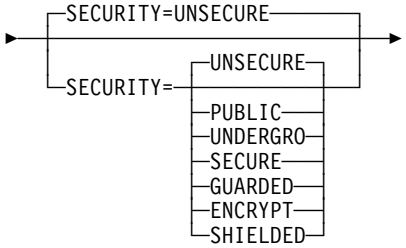
statements: PORT, PU
dependency: none
range: 4–252

On PORT: specifies the service access point (SAP) address for the connection to the LAN that this major node defines.

On PU: specifies the SAP address of the 37x5 communication controller or the processor with an IBM Token-Ring Subsystem Controller or a multi-protocol communication subsystem that this PU definition statement defines.

For the PORT or PU definition statement, the value of SAPADDR must be a multiple of 4.

SECURITY (VM)



statements: PORT
dependency: 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.

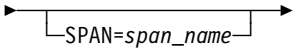
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN

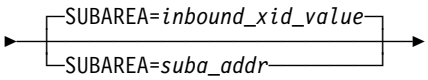


statements: GROUP, LINE, PU
dependency: NetView

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

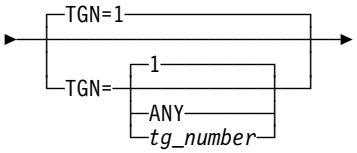


statements: PU
dependency: applies only to subarea nodes (PUTYPE=4 or 5)
range: 1—value of MXSUBNUM start option

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 received on the XID with the value you specify. If the values do not match, the connection is terminated.

The value of SUBAREA must be the same as the subarea address defined for this physical unit in the adjacent domain.

TGN



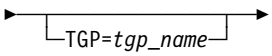
statements: PU (subarea node)
dependency: none
range: 1–255

specifies the transmission group (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 transmission group 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.

TGP

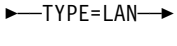


statements: PORT
dependency: APPN only

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 495 for more information on transmission group profiles.

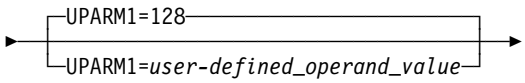
TYPE



statements: VBUILD
dependency: none

specifies that this VBUILD definition statement defines a local area network major node to VTAM. This operand is required.

UPARM1

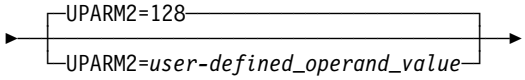


statements: PORT
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, "Forcing an APPN Route in a VTAM Network" in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

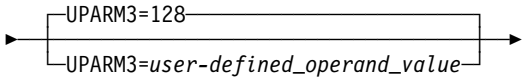
UPARM2



statements: PORT
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

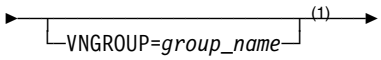
UPARM3



statements: PORT
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

VNGROUP



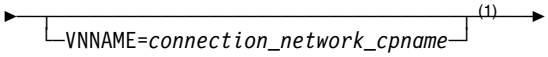
Note:
[1] Code VNGROUP if you code VNNAME.

statements: PORT
dependency: VNNAME; DIAL=YES on GROUP

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



Note:
[1] Code VNNAME if you code VNGROUP.

statements: PORT
dependency: VNGROUP; DIAL=YES 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. 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.

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 29. Local Non-SNA Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	LBUILD	CONFGDS	205				MVS,VSE
		CONFGPW	205				MVS,VSE CONFGDS
name	LOCAL	CUADDR	205	R			
		TERM	211	R			
		ASLENT	204				ASLTAB
		ASLTAB	205				
		DLOGMOD	206				MODETAB
		FEATUR2	206			(DUALCSE, NOEDATS, MODEL1, NOPRINTR, NOSELPEN)	
		ISTATUS	207			ACTIVE	
		LOGAPPL	208				NetView or TSO (MVS)
		LOGTAB	208				dependent LUs
		MDLENT	208				MDLTAB
		MDLTAB	209				
		MODETAB	209			ISTINCLM	
		REGISTER	210			NETSRVR	APPN only
		SPAN	210				NetView
		SSCPFM	210			USS3270	
		USSTAB	211			ISTINCDT	Dependent LUs

Legend:

R Required operand.

Coding Notes

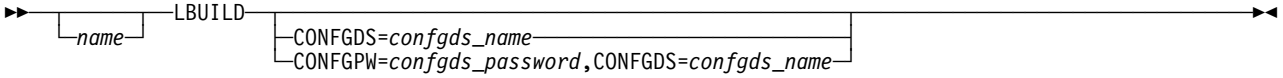
LBUILD Definition Statement: Code one LBUILD definition statement for each logical group of channel-attached non-SNA terminals.

LOCAL Definition Statement: Code one LOCAL definition statement for each channel-attached non-SNA terminal in the VTAM domain.

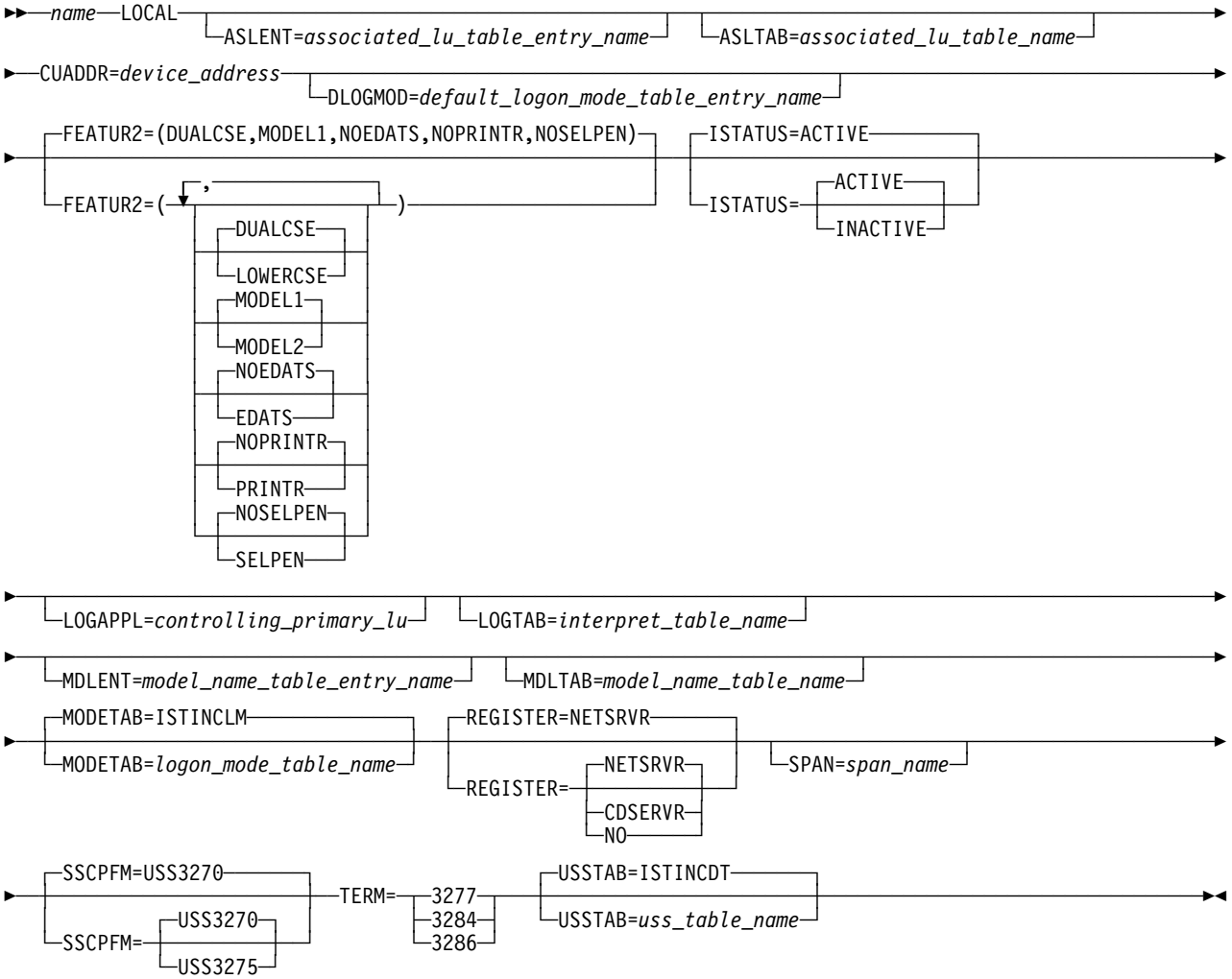
Full Syntax

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

LBUILD



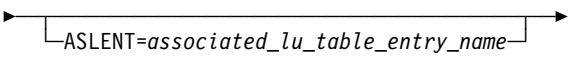
LOCAL



Operand Descriptions

The following section describes the operands you can code for the local non-SNA major node.

ASLENT

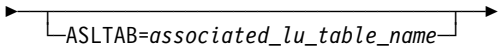


statements: LOCAL
dependency: ASLTAB

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



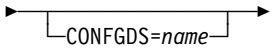
statements: LOCAL
dependency: none

specifies the name of the associated LU table to be used for the logical unit. See "Associated LU Table" on page 638 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 714 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 "MODIFY TABLE Command" in *VTAM Operation* for more information about the MODIFY TABLE command.

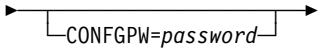
CONFGDS (MVS, VSE)



statements: LBUILD
dependency: 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 (MVS, VSE)

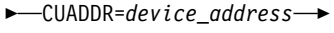


statements: LBUILD
dependency: CONFGDS

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

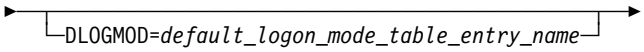


statements: LOCAL
dependency: none
range: X'000'–X'FFFF'

defines the 3-digit or 4-digit (MVS) hexadecimal channel unit address 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: MVS A local non-SNA terminal should not be defined to and activated by VTAM if its channel unit address is defined as an MVS console and allocated to console services. Activating a local non-SNA terminal whose channel unit address is in use by console services can cause VTAM, console services, or both to abend.

DLOGMOD

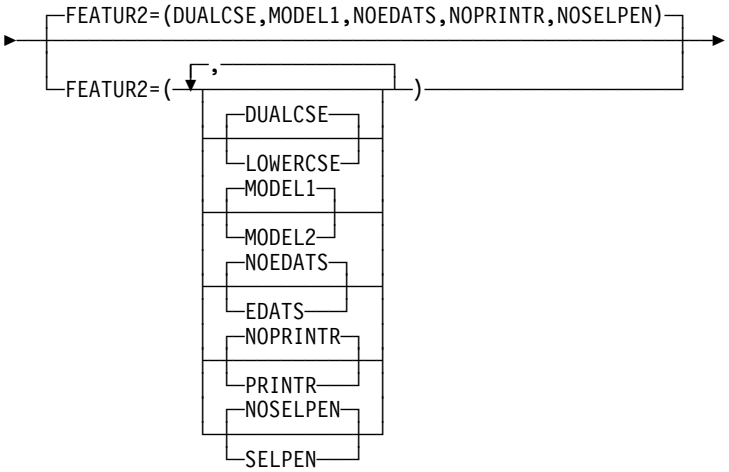


statements: LOCAL
dependency: MODETAB

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 666.

FEATUR2



statements: LOCAL
dependency: 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, “INQUIRE-Obtain Logical Unit Information or Application Program Status” in *VTAM Programming*.

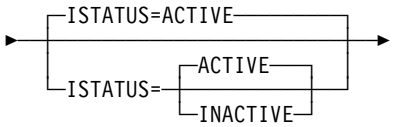
FEATUR2=NOPRINTR
FEATUR2=PRINTR
specifies whether the local device is a printer.

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 529.

FEATUR2=NOSELPEN
FEATUR2=SELPEN
specifies whether this terminal supports the use of a selector pen.

ISTATUS



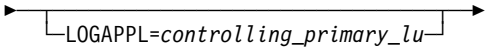
statements: LOCAL
dependency: none

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 “VARY ACT Command” in *VTAM 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



statements: LOCAL
dependency: NetView or TSO (MVS)

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.

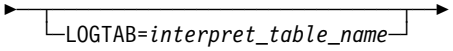
To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

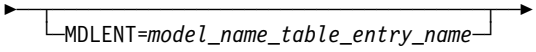
LOGTAB



statements: LOCAL
dependency: 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 659 for a description of how to define an interpret table.

MDLENT

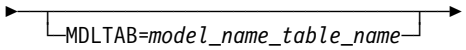


statements: LOCAL
dependency: MDLTAB

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



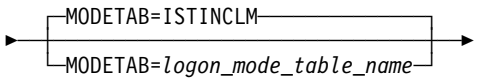
statements: LOCAL
dependency: none

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information on the MODIFY TABLE command.

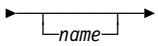
MODETAB



statements: LOCAL
dependency: 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 666. 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



statements: LBUILD

statements: LOCAL

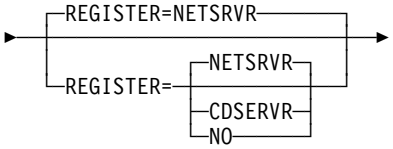
assigns a name to the resource being defined by the definition statement.

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



statements: LOCAL
dependency: APPN only

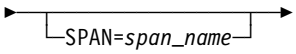
specifies how a resource should be registered.

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

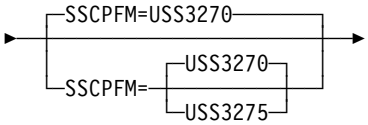


statements: LOCAL
dependency: NetView

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



statements: LOCAL
dependency: none

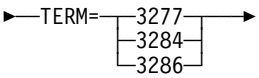
determines what type of RUs the logical unit or terminal can support in its communications with the SSCP.

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.

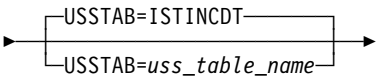
TERM



statements: LOCAL
dependency: none

specifies a specific, channel-attached non-SNA terminal (printer or display station). You can code only 3277, 3284, or 3286.

USSTAB



statements: LOCAL
dependency: 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 690.

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 “Channel Connections” in the *VTAM Network Implementation Guide* for a discussion of requirements for defining SNA and non-SNA devices.

Table 30 (Page 1 of 2). Local SNA Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LOCAL	235	R			
		CONFGDS	218				MVS,VSE
		CONFGPW	218				MVS,VSE CONFGDS
name	PU	AUTHLEN	217			YES	APPN only PU Type 2.1
		CAPACITY	217			8K	APPN only
		CONNTYPE	218			CONNTYPE start option	APPN only; NODETYPE
		COSTBYTE	219			0	APPN only
		COSTTIME	219			0	APPN only
		CPCP	219			CPCP start option	APPN only
		CPNAME	220				
		CUADDR	220				TRLE
		DELAY	220			0.200	
		DISCNT	220			(NO)	
		DYNADJCP	221			DYNADJCP start option	APPN only; CPNAME
		DYNLU	222			DYNLU start option	
		ISTATUS	224			ACTIVE	
		LIMRES	224			NO	LU 6.2 only
		LUGROUP	227				
		LUSEED	227				LUGROUP
		MAXBFRU	227			1	TRLE
		NATIVE	229				BN=YES APPN only
		NETID	230				
		NN	230				APPN only
		PDELAY	231			TERRESTR	APPN only
		PUTYPE	231			2	
		SECNET	232			NO	3174 token ring
		SECURITY	232			UNSECURE	APPN only MVS,VM NetView
		SPAN	233				
		TGN	234			ANY	APPN only
		TGP	234				APPN only
		TOPO	234			CONNECT	APPN only
		TRLE	235				APPN only CUADDR
		UPARM1	235			128	APPN only
		UPARM2	235			128	APPN only
		UPARM3	236			128	APPN only
		VERALSID	236			NO	APPN only
		XID	237			NO	

Table 30 (Page 2 of 2). Local SNA Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	LU	LOCADDR	225	R			see description
		ASLENT	216		P		ASLTAB
		ASLTAB	216		P		
		CLRSESSQ	217		P	NO	
		DLOGMOD	221		P		MODETAB
		EAS	222		P	256	independent LUs only
		ENCR	223		P	NONE	MVS
		FASTPASS	223		P	YES	TSO only
		ISTATUS	224			ACTIVE	
		LOGAPPL	225		P		NetView or TSO (MVS)
		LOGTAB	226		P		dependent LUs
		LUDR	226			YES	
		MDLENT	228		P		MDLTAB
		MDLTAB	228		P		
		MODETAB	228		P	ISTINCLM	
		PACING	230		P	1	
		REGISTER	231		P	see description	APPN only
		SPAN	233				NetView
		SSCPFM	233		P	FSS	
		USSTAB	236		P	ISTINCDT	dependent LUs
		VPACING	236		P	1	

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	MDLENT
ASLTAB	MDLTAB
DLOGMOD	MODETAB
ENCR	PACING
FASTPASS	REGISTER
ISTATUS *	SSCPFM
LOCADDR	USSTAB
LOGAPPL	VPACING
LOGTAB	

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 “Dynamic Reconfiguration and Change of Operands” in the *VTAM Network Implementation Guide* and “VARY ACT Command” in *VTAM Operation* for more information on the VARY ACT,UPDATE=ALL command.

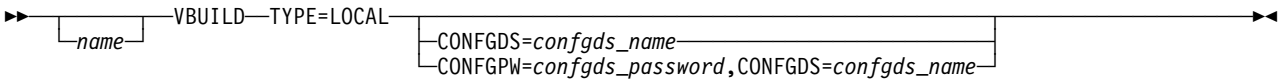
NODETYPE Start Option: Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

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.

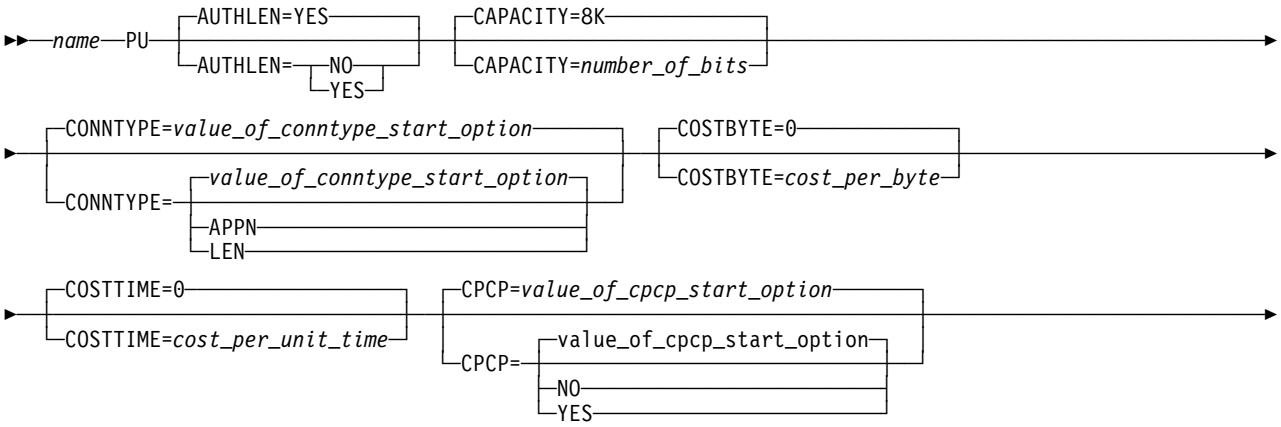
Full Syntax

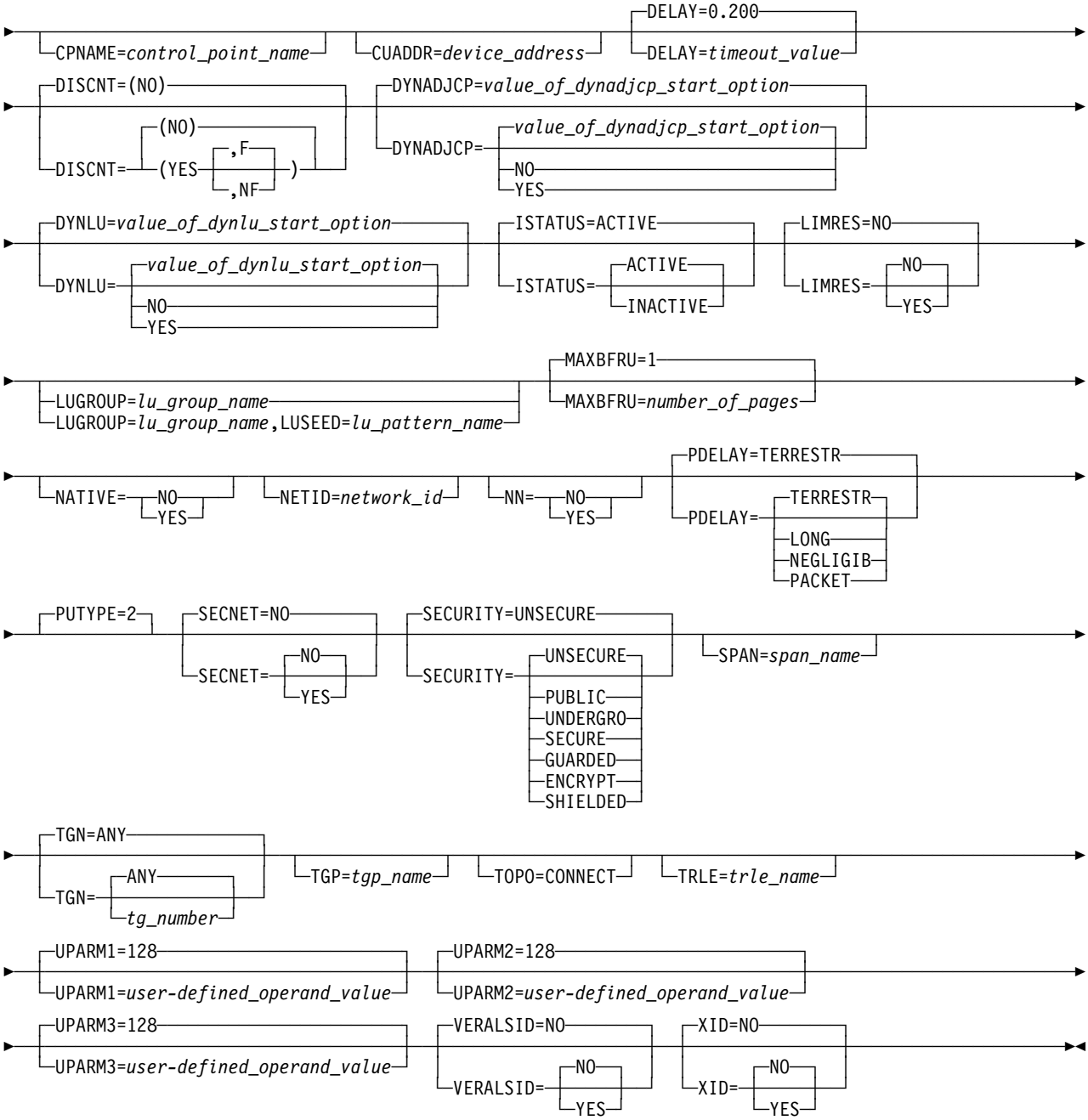
The full syntax for the local SNA major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

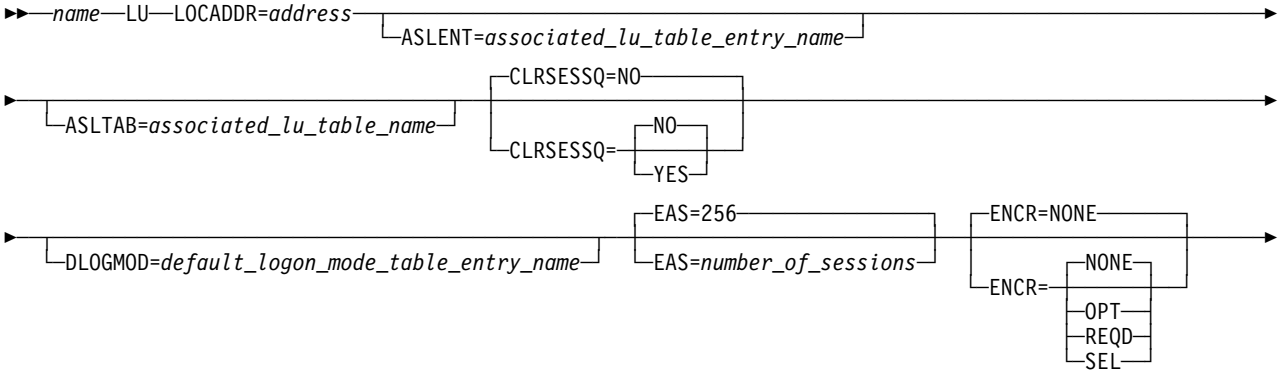


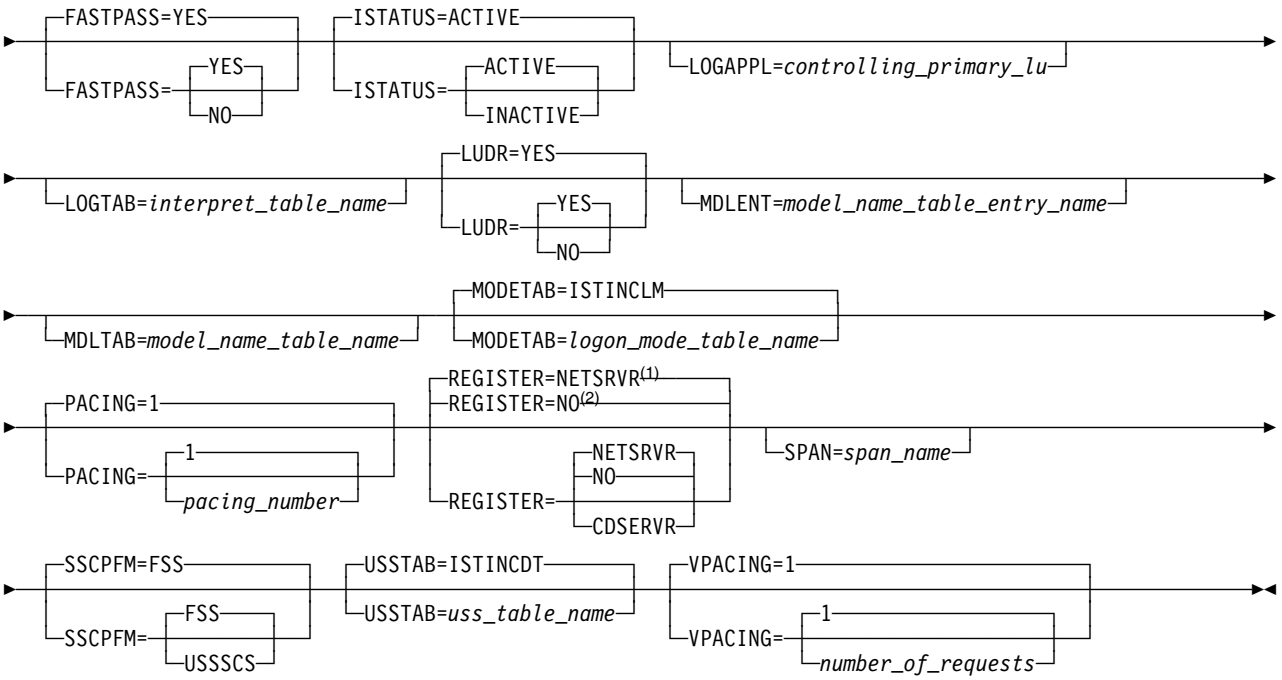
PU




CPNAME=control_point_name
CUADDR=device_address
DELAY=0.200
DELAY=timeout_value
DISCNT=(NO)
DISCNT=
(NO)
(YES
,F
,NF
)
DYNADJCP=value_of_dynadjcp_start_option
DYNADJCP=
value_of_dynadjcp_start_option
NO
YES
DYNLU=value_of_dynlu_start_option
DYNLU=
value_of_dynlu_start_option
NO
YES
ISTATUS=ACTIVE
ISTATUS=
ACTIVE
INACTIVE
LIMRES=NO
LIMRES=
NO
YES
LUGROUP=lu_group_name
LUGROUP=lu_group_name,LUSEED=lu_pattern_name
MAXBFRU=1
MAXBFRU=number_of_pages
NATIVE=
NO
YES
NETID=network_id
NN=
NO
YES
PDELAY=TERRESTR
PDELAY=
TERRESTR
LONG
NEGLIGIB
PACKET
PUTYPE=2
SECNET=NO
SECNET=
NO
YES
SECURITY=UNSECURE
SECURITY=
UNSECURE
PUBLIC
UNDERGRO
SECURE
GUARDED
ENCRYPT
SHIELDED
SPAN=span_name
TGN=ANY
TGN=
ANY
tg_number
TGP=tgp_name
TOPO=CONNECT
TRLE=trle_name
UPARM1=128
UPARM1=user-defined_operand_value
UPARM2=128
UPARM2=user-defined_operand_value
UPARM3=128
UPARM3=user-defined_operand_value
VERALSID=NO
VERALSID=
NO
YES
XID=NO
XID=
NO
YES

LU


name
LU
LOCADDR=address
ASLENT=associated_lu_table_entry_name
ASLTAB=associated_lu_table_name
CLRSESSQ=NO
CLRSESSQ=
NO
YES
DLOGMOD=default_logon_mode_table_entry_name
EAS=256
EAS=number_of_sessions
ENCR=NONE
ENCR=
NONE
OPT
REQD
SEL



Notes:

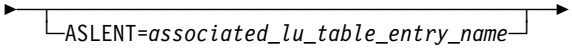
1 REGISTER defaults to NETSRVR for dependent LUs.

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

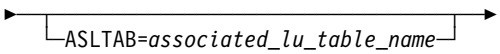


statements: PU, LU
dependency: ASLTAB

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



statements: PU, LU
dependency: none

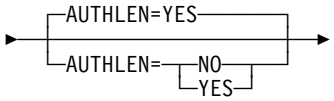
specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 638 for a description of how to define an associated LU table.

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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information on the MODIFY TABLE command.

AUTHLEN



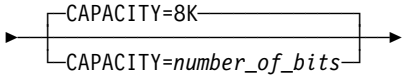
statements: PU
dependency: PU Type 2.1; APPN only

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

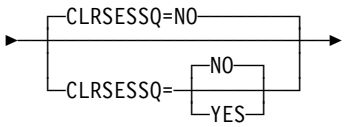


statements: PU
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CLRSESSQ



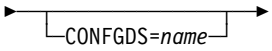
statements: PU, LU
dependency: none

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.

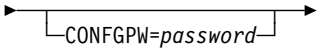
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

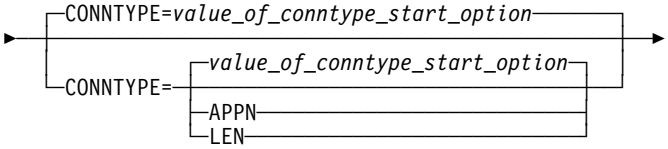


statements: VBUILD
dependency: CONFGDS

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



statements: PU
dependency: NODETYPE; APPN only

specifies for a PU 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.

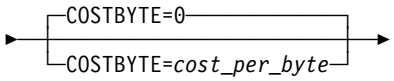
When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

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.

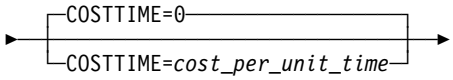
COSTBYTE



statements: PU
dependency: 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.

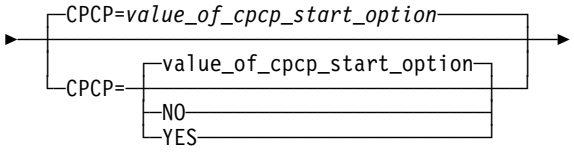
COSTTIME



statements: PU
dependency: 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



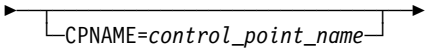
statements: PU
dependency: APPN only

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.

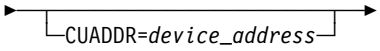
CPNAME



statements: PU
dependency: 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



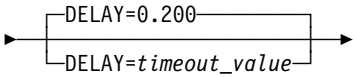
statements: PU
dependency: TRLE
range: X'000'–X'FFFF'

defines a 3-digit or 4-digit (MVS) 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

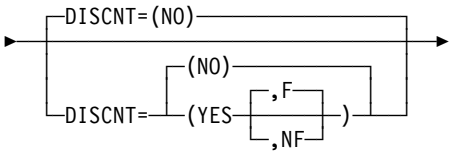


statements: PU
dependency: none
range: 0–9.999

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, set this value to 0. Refer to “DELAY Operand” in the *VTAM Network Implementation Guide* for more information.

DISCNT



statements: PU
dependency: 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 "Switched SDLC Connection" in the *VTAM Network Implementation Guide* for a discussion of conditions that can cause sessions to terminate.

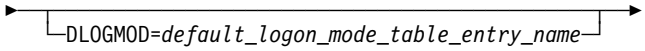
DISCNT=(YES)
If you code YES, VTAM automatically terminates the SSCP-LU and SSCP-PU sessions as soon as the last LU-LU session is terminated.

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

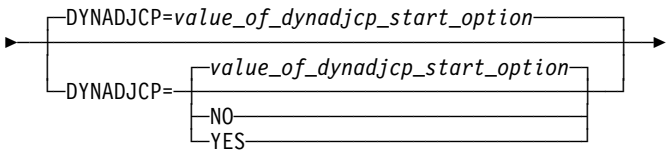


statements: PU, LU
dependency: MODETAB

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 666.

DYNADJCP



statements: PU
dependency: APPN only; ignored if CPNAME coded

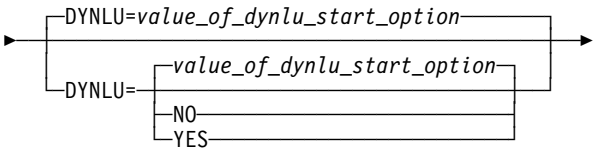
specifies whether an ADJCP can be created dynamically for an adjacent node.

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



statements: PU
dependency: 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.

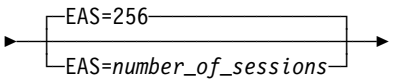
DYNLU=YES
specifies that dynamic allocation of CDRSC definitions is supported.

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 23 under the adjacent control point major node.

EAS



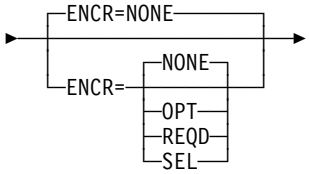
statements: PU, LU
dependency: LOCADDR=0
range: 0–65535

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 534 for information about BSBUF.

ENCR (MVS)



statements: PU, LU
dependency: none

specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

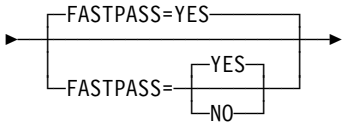
ENCR=NONE
specifies that this logical unit cannot engage in cryptographic sessions.

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.

FASTPASS (MVS)



statements: PU, LU
dependency: 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.

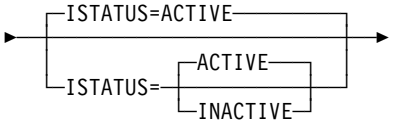
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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.

ISTATUS



statements: PU, LU
dependency: none

determines whether a physical unit or logical unit is to be activated after the first activation of the channel-attachment major node.

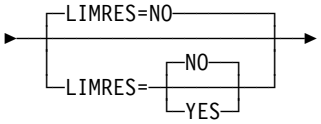
ISTATUS=ACTIVE
specifies that the physical unit or logical unit becomes active when its channel becomes active.

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 "VARY ACT Command" in *VTAM Operation* for information on using the VARY ACT command.

Note: You can dynamically change ISTATUS only on the LU definition statement.

LIMRES



statements: PU
dependency: LU 6.2 only

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.

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.

LIMRES applies to LU 6.2 conversations only.

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

statements: LU
dependency: see notes
range: 0–63 (PU type 1)
0–255 (PU type 2)

specifies the logical unit's local address at the physical unit.

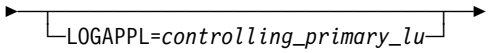
For independent logical units, specify 0. For dependent logical units, specify a non-zero value. All dependent logical units under a given physical unit must have unique LOCADDR values.

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



statements: PU, LU
dependency: NetView or TSO (MVS)

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.

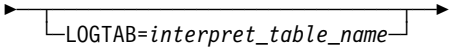
To ensure retry of a pending automatic logon, code the AUTORTRY or AUTOTI start options.

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

LOGTAB

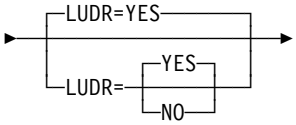


statements: PU, LU
dependency: dependent LUs only

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 659 for a description of how to define an interpret table.

LOGTAB is valid only for dependent logical units.

LUDR



statements: LU
dependency: none

specifies whether an LU is eligible for dynamic reconfiguration.

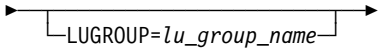
This operand is ignored for independent LUs.

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.

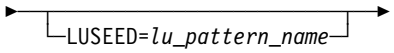
LUGROUP



statements: PU
dependency: 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



statements: PU
dependency: LUGROUP

provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

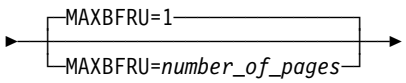
If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- 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 “Selection of Definitions for Dependent LUs (SDDLU) Exit Routine” in *VTAM Customization*.

This operand is valid only when you code the LUGROUP operand.

MAXBFRU



statements: PU
dependency: 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.

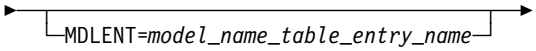
number_of_buffers is a decimal integer. Always code MAXBFRU such that the *bufsize* value for the IOBUF buffer pool times MAXBFRU is equal to or greater than the maximum RU size used by the controller.

Note: Do not code MAXBFRU=1 or use the default for a Model 3174.

For more information on determining a value for MAXBFRU, refer to “Guidelines for Setting MAXBFRU” in the *VTAM 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.

MDLENT

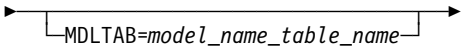


statements: PU, LU
dependency: MDLTAB

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



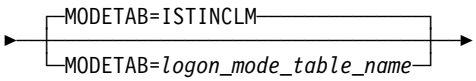
statements: PU, LU
dependency: none

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information on the MODIFY TABLE command.

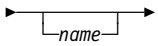
MODETAB



statements: PU, LU
dependency: 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 666. 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



statements: VBUILD

statements: PU, LU

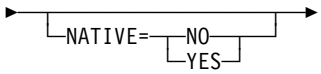
On VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

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



statements: PU
dependency: 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.

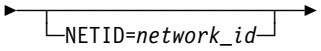
- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

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



statements: PU
dependency: none

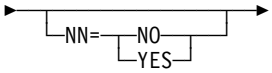
specifies a 1–8 character network identifier.

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.

This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See page 600 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. 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



statements: PU
dependency: APPN only

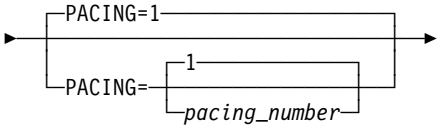
specifies whether the adjacent node is expected to be a network node.

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

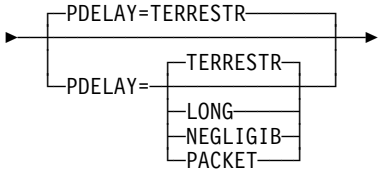


statements: PU, LU
dependency: none
range: 0–63

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

PDELAY



statements: PU
dependency: 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).

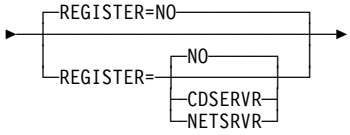
PUTYPE

statements: PU
dependency: none

specifies the physical unit type. PUTYPE=2 is the only valid value.

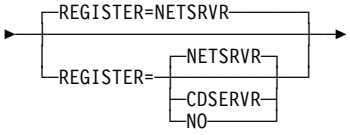
REGISTER

For Independent LUs:



statements: PU, LU
dependency: APPN only

For Dependent LUs:



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.

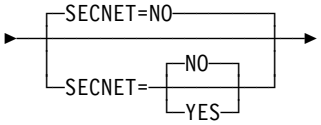
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

SECNET

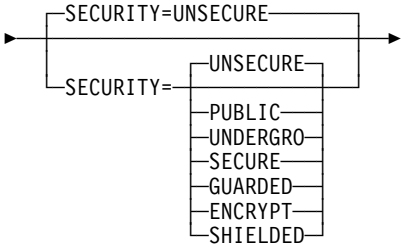


statements: PU
dependency: valid only for 3174 token-ring attachments

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 (MVS, VM)



statements: PU
dependency: 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.

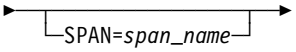
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN

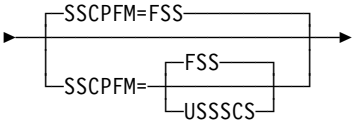


statements: PU, LU
dependency: NetView

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



statements: PU, LU
dependency: none

determines what type of RUs the logical unit can support in its communications with the SSCP.

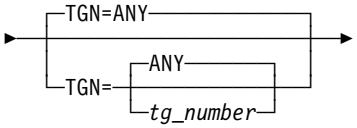
SSCPFM=FSS
specifies that the logical unit supports formatted commands in its communication with the SSCP.

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



statements: PU
dependency: APPN only
range: 0–21

specifies the transmission group (TG) number that represents the connection between this node and the destination APPN node.

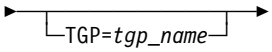
By specifying a *tg_number*, you indicate a preferred TG that might be used depending on how the partner nodes negotiate the number.

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

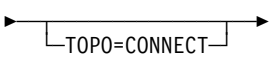


statements: PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO



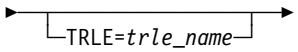
statements: PU
dependency: APPN only

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



statements: PU
dependency: CUADDR; APPN only

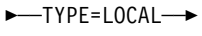
specifies the 8-character name of the TRLE definition statement to be used for the physical unit being defined by this PU definition statement. The TRLE determines the transport characteristics of the PU.

Each TRLE definition statement can be used by only one PU. If you code the same TRLE name on more than one PU, activation for the second PU fails.

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.

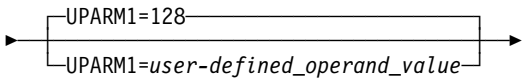
TYPE



statements: VBUILD
dependency: none

specifies a local SNA major node. This operand is required.

UPARM1

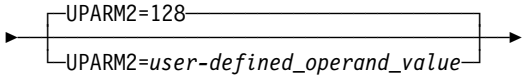


statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

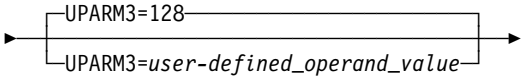
UPARM2



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

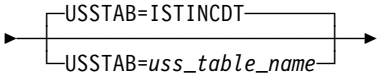
UPARM3



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

USSTAB



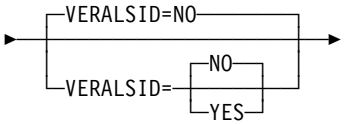
statements: PU, LU
dependency: dependent LUs

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 690.

USSTAB is valid for dependent logical units only.

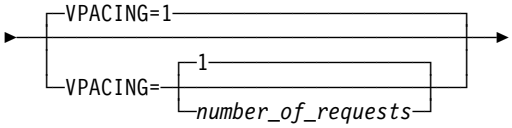
VERALSID



statements: PU
dependency: 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

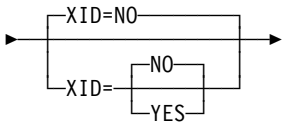


statements: PU, LU
dependency: none
range: 0–63

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

XID



statements: PU
dependency: 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 PU type 2.1 channel contact procedure is to be used.

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 31. LU Group Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=LUGROUP	246	R			
name	LUGROUP						
name	LU	ASLENT	240				ASLTAB
		ASLTAB	240				
		DLOGMOD	240				MODETAB
		ENCR	241			NONE	MVS
		FASTPASS	241			YES	TSO only
		LOGAPPL	242				NetView or TSO (MVS) dependent LUs
		LOGTAB	242				
		LUDR	243			YES	
		MDLENT	243				MDLTAB
		MDLTAB	243				
		MODETAB	244			ISTINCLM	
		PACING	245			1	
		REGISTER	245			NETSRVR	APPN only
		SSCPFM	246			FSS	
		USSTAB	246			ISTINCDT	dependent LUs
		VPACING	247			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.

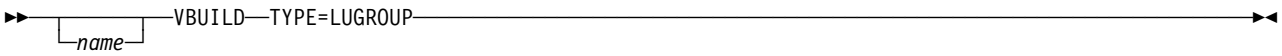
LU Definition Statement: Code an LU definition statement for each model LU in the LU group. On type 2.0 and type 2.1 devices that support dynamic definition of dependent LUs, the model LU definitions are used to dynamically define dependent LUs when the device powers on. The maximum number of model LUs that you can define for each LU group is 255.

You cannot take advantage of the sift-down effect in this major node.

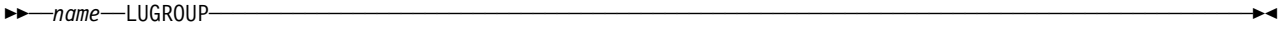
Full Syntax

The full syntax for the LU group major node follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

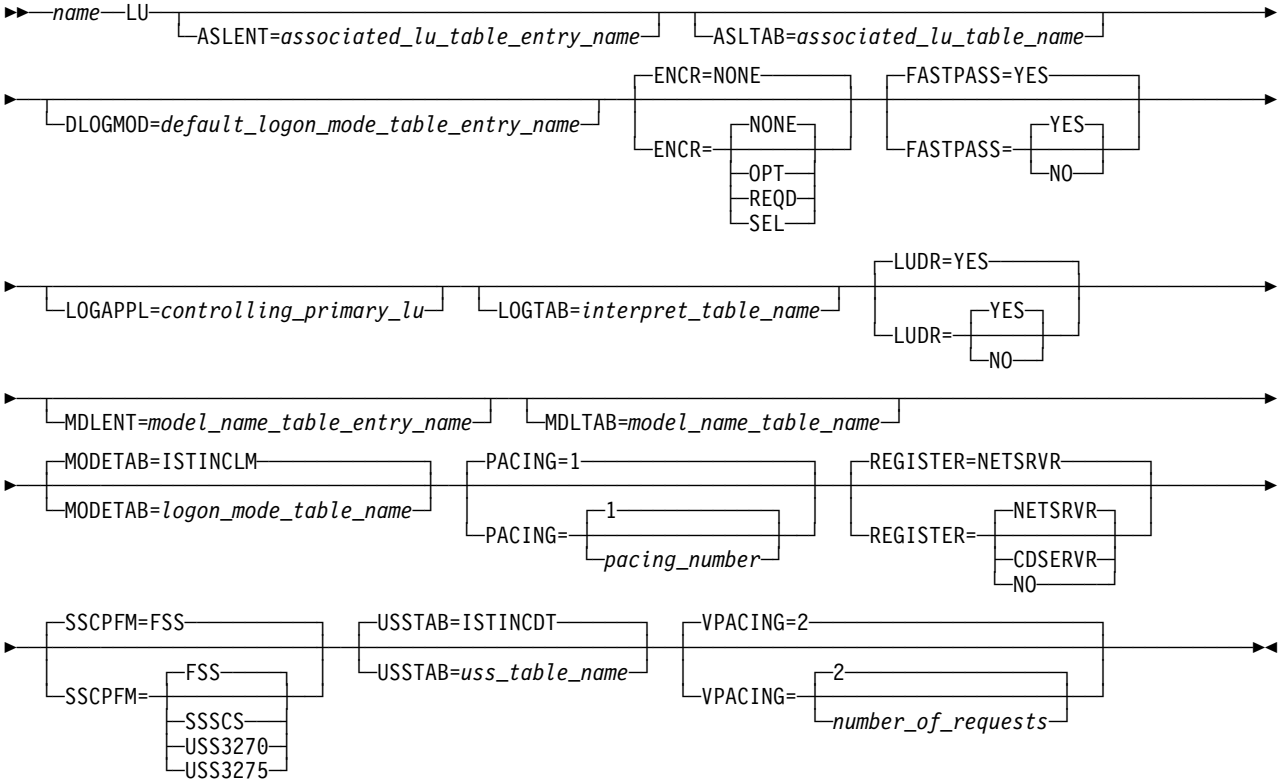
VBUILD



LUGROUP



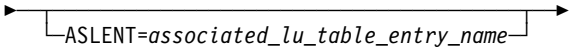
LU



Operand Descriptions

The following section describes the operands you can code for the LU group major node.

ASLENT

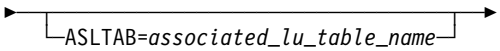


statements: LU
dependency: ASLTAB

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



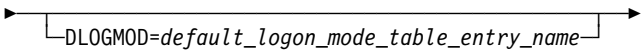
statements: LU
dependency: none

specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 638 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

DLOGMOD

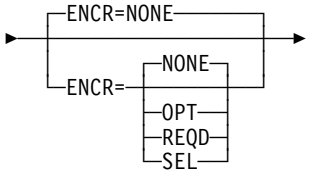


statements: LU
dependency: MODETAB

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 666.

ENCR (MVS)



statements: LU
dependency: none

tells whether this logical unit has any special requirements for enciphering and deciphering messages.

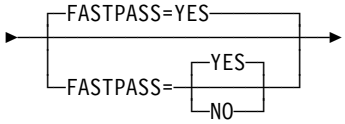
ENCR=NONE
specifies that this logical unit cannot engage in cryptographic sessions.

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.

FASTPASS (MVS)



statements: LU
dependency: 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.

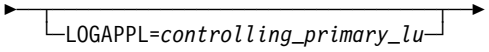
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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



statements: LU
dependency: NetView or TSO (MVS)

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.

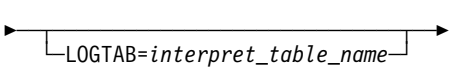
You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

LOGTAB

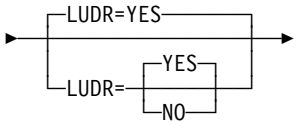


statements: LU
dependency: valid only for dependent LUs

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 659 for a description of how to define an interpret table.

LOGTAB is valid for dependent logical units only.

LUDR



statements: LU
dependency: none

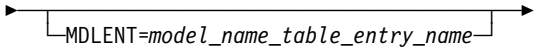
specifies whether an LU is eligible for dynamic reconfiguration.

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

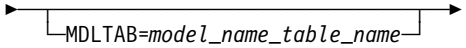


statements: LU
dependency: MDLTAB

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



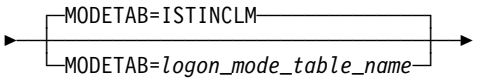
statements: LU
dependency: none

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation*. for more information about the MODIFY TABLE command.

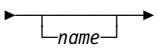
MODETAB



statements: LU
dependency: 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 666. 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



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 “Defining Dependent LUs Dynamically” in the *VTAM Network Implementation Guide*.

The following are valid characters:

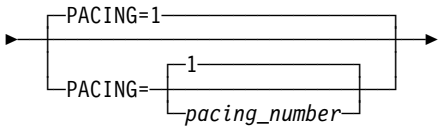
- Alphabetical (A—Z)
- Numerical (0–9)
- National (@, #, or $).

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

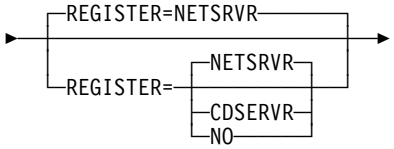


statements: LU
dependency: none
range: 0–63

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

REGISTER



statements: LU
dependency: 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.

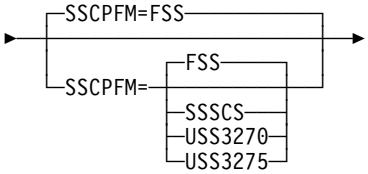
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

SSCPFM



statements: LU
dependency: none

determines what type of RUs the logical unit can support in its communications with the SSCP.

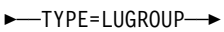
SSCPFM=FSS
specifies that field-formatted messages are supported.

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.

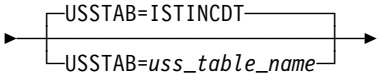
TYPE



statements: VBUILD
dependency: 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



statements: LU
dependency: dependent LUs only

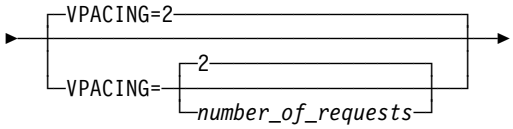
specifies the name of a USS table that VTAM uses to process character-coded input that it receives from the logical unit.

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 690.

USSTAB is valid for dependent logical units only.

VPACING



statements: LU
dependency: 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 “Session Pacing” in the *VTAM 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 32 (Page 1 of 2). Model Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=MODEL	277	R			
name	PU	ADDR	253	R			nonswitched connections
		ANS	253			STOP	
		AUTHLEN	254			YES	APPN only PU Type 2.1
		CAPACITY	255			8K	APPN only
		CONNTYPE	255			CONNTYPE start option	APPN only
		COSTBYTE	256			0	APPN only
		COSTTIME	256			0	APPN only
		CPCP	256			CPCP start option	APPN only
		DATMODE	257			HALF	
		DISCNT	257			(NO)	
		DYNADJCP	258			DYNADJCP start option	APPN only
		DYNLU	259			DYNLU start option	CDRDYN
		IRETRY	261			NO	
		LANACK	262			(0,0)	VM,VSE
		LANCON	262				VM,VSE
		LANINACT	263			4.8	VM,VSE
		LANRESP	263				VM,VSE
		LANSDWDW	264			(2,1)	VM,VSE
		LANSW	264			NO	VM,VSE
		LIMRES	265			NO	LU 6.2 only
		LUGROUP	267				
		LUSEED	267				LUGROUP
		MACADDR	267				VM,VSE
		MAXDATA	268			261 or 265	
		MAXOUT	268			1	
		MAXSESS	269				NCP-attached independent LUs
		NATIVE	270				BN=YES APPN only
		NETID	271				
		NN	271				APPN only
		PASSLIM	272			1	
		PDELAY	272			TERRESTR	APPN only
		PUTYPE	273			2	
		SAPADDR	274			4	VM,VSE
		SECNET	274			NO	
		SECURITY	275			UNSECURE	APPN only MVS,VM

Table 32 (Page 2 of 2). Model Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		SPAN	275				NetView
		TGP	277				APPN only
		TOPO	277			CONNECT	APPN only
		UPARM1	277			128	APPN only
		UPARM2	278			128	APPN only
		UPARM3	278			128	APPN only
		VERALSID	278			NO	APPN only
name	LU	LOCADDR	265	R			see description
		ASLENT	254				ASLTAB
		ASLTAB	254				
		BATCH	255			NO	Pre-V4R3 NCP only
		DLOGMOD	258				MODETAB
		ENCR	259			NONE	MVS
		FASTPASS	260			YES	TSO only
		FEATUR2	260			(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	
		LOGAPPL	266				NetView or TSO (MVS)
		LOGTAB	266				
		MDLENT	269				MDLTAB
		MDLTAB	269				
		MODETAB	270			ISTINCLM	
		PACING	272			1	
		REGISTER	273			see description	
		SPAN	275				NetView
		SSCPFM	275			FSS	
		TERM	276				
		USSTAB	278			ISTINCDT	dependent LUs
		VPACING	279			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 Appendix D, “Sample Configuration Services XID Exit Routine” in *VTAM 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.

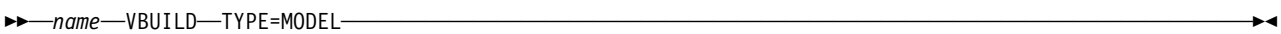
NODETYPE Start Option: Note that if you do not code the NODETYPE start option, any APPN operands you code in this major node are ignored.

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.

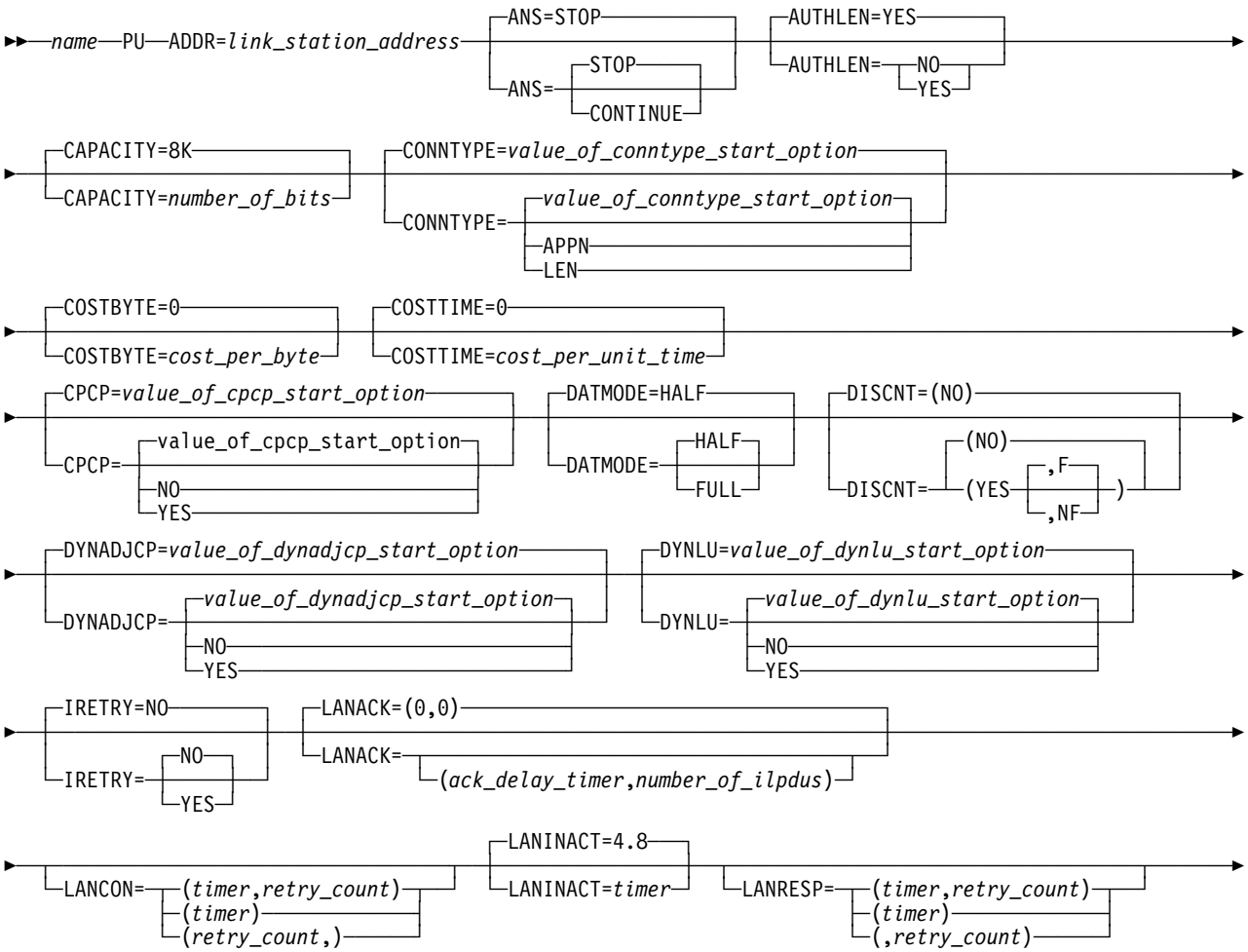
Full Syntax

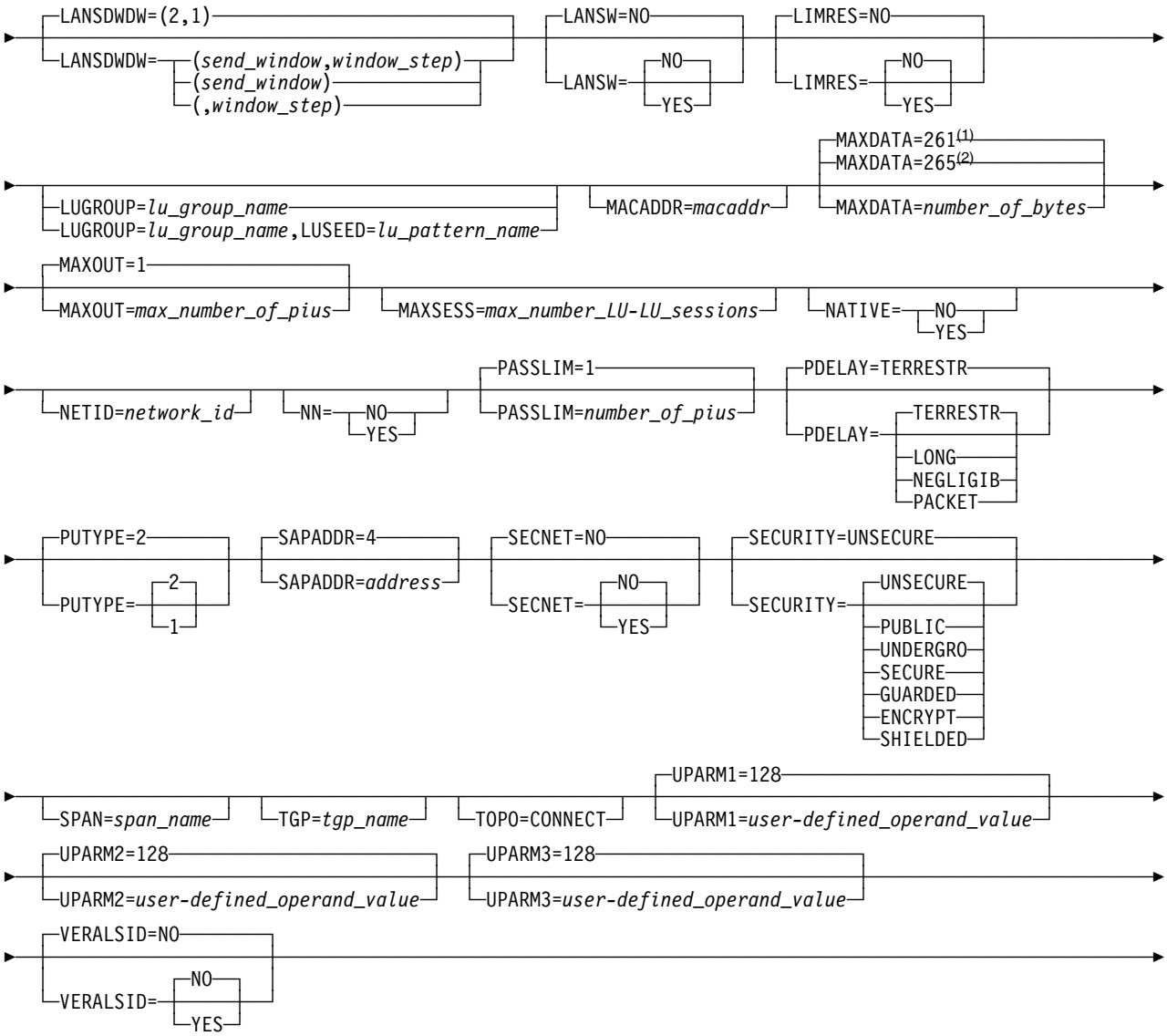
The full syntax for the model major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD



PU



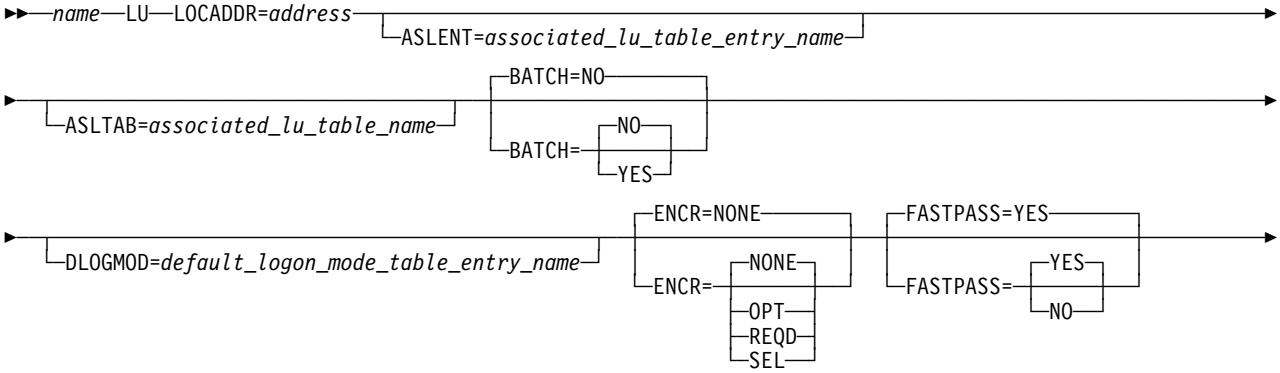


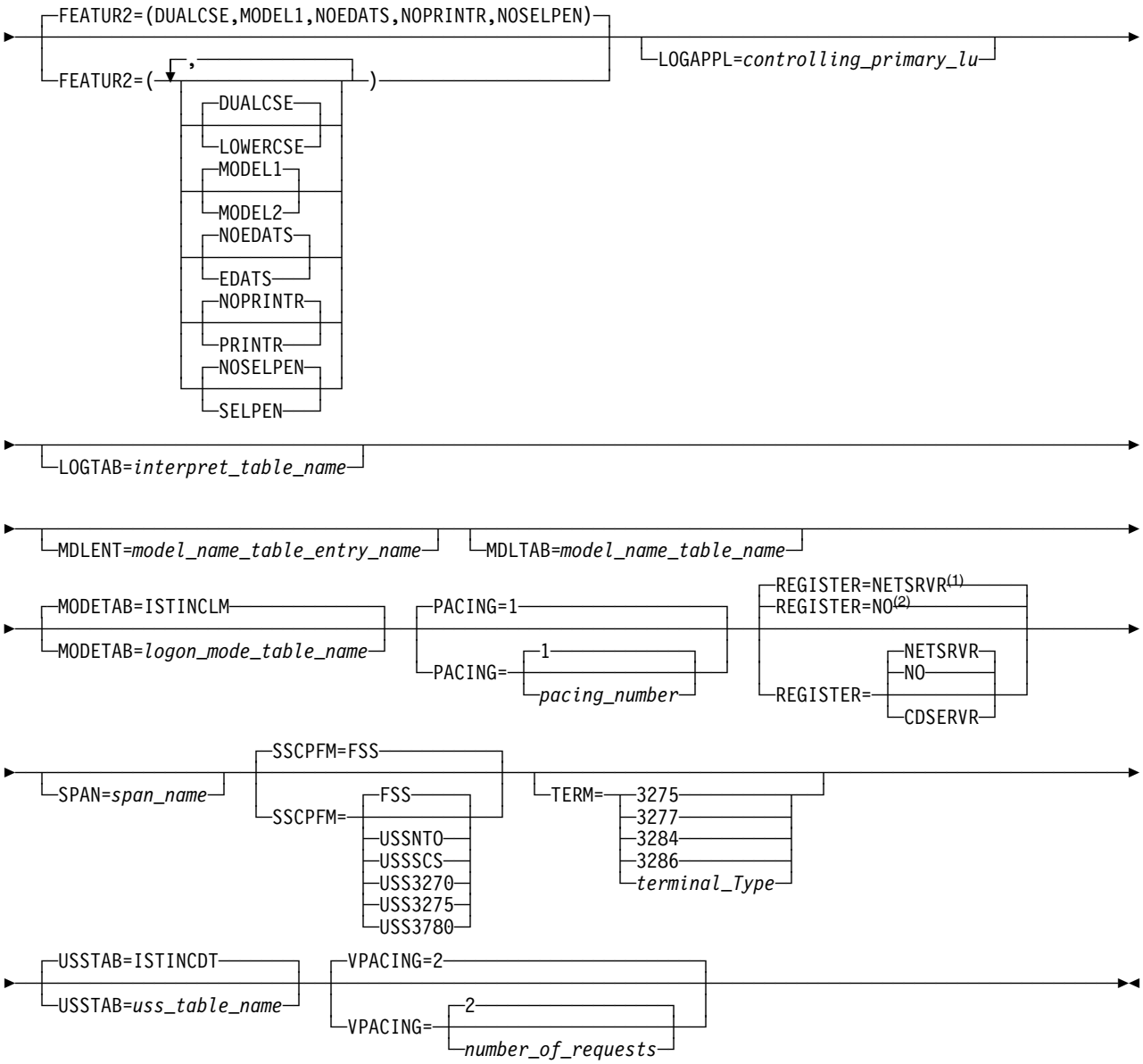
Notes:

1 MAXDATA defaults to 261 when PUTYPE=1 is coded.

2 MAXDATA defaults to 265 when PUTYPE=2 is coded.

LU





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 model major node.

ADDR

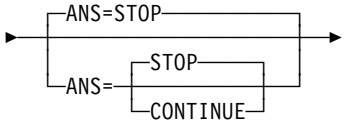
statements: PU
dependency: nonswitched connections

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



statements: PU
dependency: 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.

ANS=CONTINUE or CONT
indicates that LU-LU sessions will continue when the NCP enters automatic network shutdown.

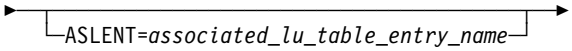
MVS You must code ANS=CONTINUE or CONT if the terminal is to participate in XRF sessions.

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 “SSCP Takeover” in the *VTAM 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

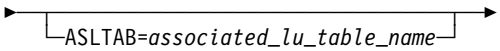


statements: LU
dependency: ASLTAB

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



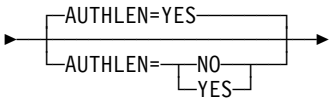
statements: LU
dependency: none

specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 638 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

AUTHLEN



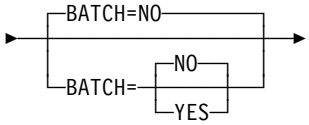
statements: PU
dependency: PU Type 2.1; APPN only

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



statements: LU
dependency: pre-V4R3 NCPs only

specifies the processing priority that the NCP uses for the logical unit.

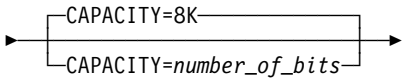
BATCH=NO
specifies a high priority (suitable for interactive application programs).

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

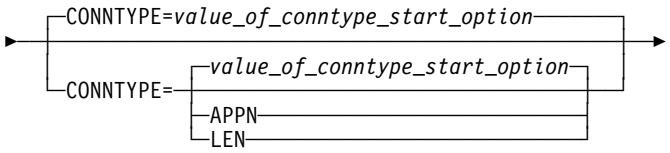


statements: PU
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CONNTYPE



statements: PU
dependency: APPN only

specifies for a PU 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.

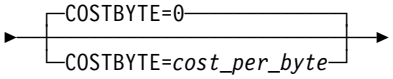
When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

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.

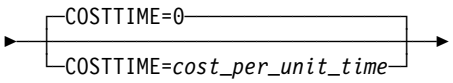
COSTBYTE



statements: PU
dependency: 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.

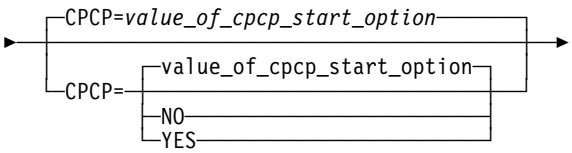
COSTTIME



statements: PU
dependency: 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



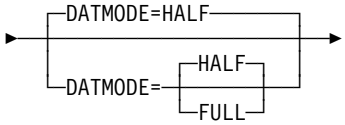
statements: PU
dependency: APPN only

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.

CPCP=NO
CP-CP sessions are not supported on this connection.

CPCP=YES
CP-CP sessions are supported on this connection.

DATMODE



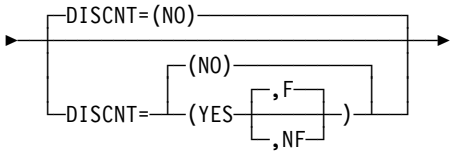
statements: PU
dependency: none

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 can communicate in full-duplex data mode.

DATMODE=HALF
indicates that the physical unit can communicate in half-duplex data mode.

DISCNT



statements: PU
dependency: 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 “Switched SDLC Connection” in the *VTAM 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.

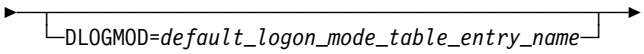
DISCNT=(YES,F)
DISCNT=(YES,NF)
specifies whether VTAM indicates final-use status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES.

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

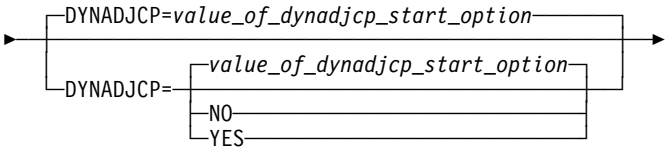


statements: LU
dependency: MODETAB

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 666.

DYNADJCP



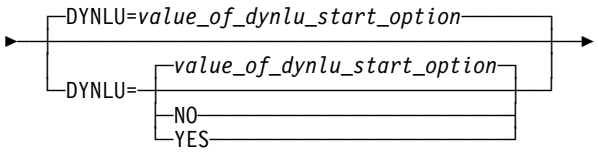
statements: PU
dependency: APPN only

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.

DYNADJCP=YES
indicates that the ADJCP can be created dynamically for the adjacent node.

DYNLU



statements: PU
dependency: 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.

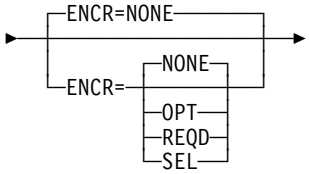
If dynamic allocation of CDRSC definitions is not supported for a particular PU, an independent logical unit that was dynamically defined using CDRSC definitions over another adjacent link station cannot use that particular PU to connect to a session.

DYNLU=YES only works if a host CDRM is active with CDRDYN=YES.

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.

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 23 under the adjacent control point major node.

ENCR (MVS)



statements: LU
dependency: none

tells whether this logical unit has any special requirements for enciphering and deciphering messages.

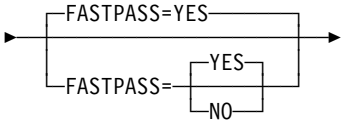
ENCR=NONE
specifies that this logical unit cannot engage in cryptographic sessions.

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.

FASTPASS (MVS)



statements: LU
dependency: 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.

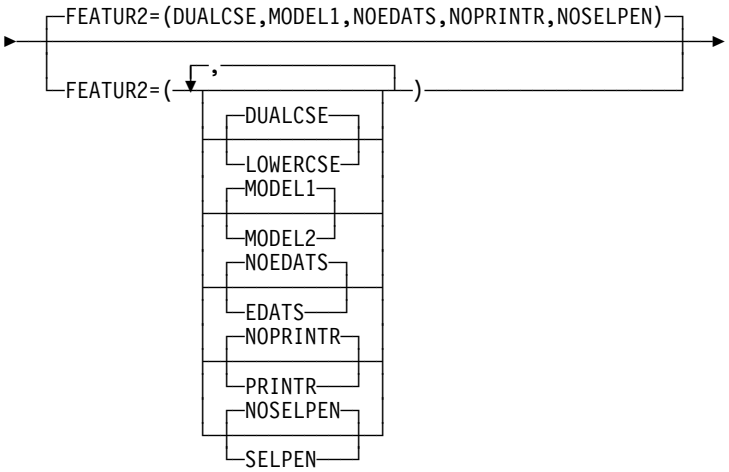
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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



statements: LU
dependency: 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.

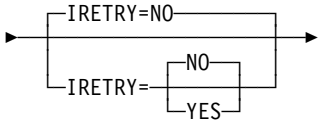
FEATUR2=MODEL1
FEATUR2=MODEL2
identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

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 “INQUIRE-Obtain Logical Unit Information or Application Program Status” in *VTAM 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.

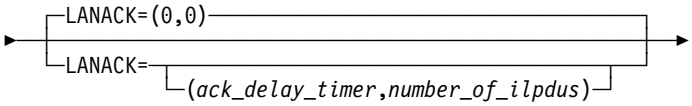
IRETRY



statements: PU
dependency: 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*

LANACK (VM, VSE)



statements: PU
dependency: see description
range: 0–25.5 (delay timer)
0–127 (ilpdus number)

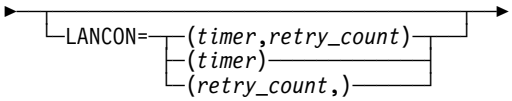
specifies the acknowledgement delay timer (t2) and number (n3) of information-LPDUs (I-LPDUs) received prior to sending an acknowledgement. If the timer expires, or if the count is reached, an acknowledgement is sent for any outstanding data.

The timer value is expressed in seconds or tenths of seconds. A value of 0 means that the timer is disabled.

You cannot specify 0 or take the default for just the timer or the counter. If you code one as 0, you must code 0 for both. For example, you cannot code LANACK=(0,2). If you want the timer to be 0, you must code LANACK=(0,0). Similarly, you cannot code LANACK=(3,0). If you want the counter to be 0, you must code LANACK=(0,0).

This operand applies only if the physical unit can be attached directly to a processor through the IBM Token-Ring Subsystem Controller, or to a processor through the multi-protocol communication subsystem.

LANCON (VM, VSE)



statements: PU
dependency: see description
range: 0–25.4 (timer)
0–254 (retry count)

specifies the timer (*ct1*) and the retry count (*cn2*) for the connection logical link control (LLC) protocol data units (LPDUs).

ct1 specifies the time, in seconds or tenths of seconds, that elapses before a retry is attempted during connection or disconnection of a link station.

If you specify 0 or do not specify a value for the timer, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

cn2 specifies the number of times to retry a transmission during connection or disconnection of a link station.

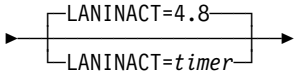
A value of 0 means that no retry will be attempted.

When the number of retries (*cn2*) has been reached and the timer (*ct1*) has elapsed for each retry, the link station is considered to be inoperative.

If you do not specify LANCON for this physical unit, the default values for the timer and the retry count are the values that you specify on the LANCON operand of the PORT definition statement in the LAN major node.

This operand applies only if the physical unit can be attached directly to a processor through the IBM Token-Ring Subsystem Controller, or to a processor through the multi-protocol communication subsystem.

LANINACT (VM, VSE)



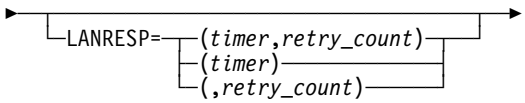
statements: PU
dependency: see description
range: 0–25.5 seconds

specifies the timer inactivity (*t*) in seconds or tenths of seconds. *t* specifies the time used to determine an inactive condition on a link station.

If you specify 0, the timer is disabled.

This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANRESP (VM, VSE)



statements: PU
dependency: see description
range: 0–25.4 seconds (timer)
0–254 (retry count)

specifies the timer (*t1*) and the retry count (*n2*) for a station in connected state.

t1 specifies the time, in seconds and tenths of seconds, that elapses before a retry is attempted when a station is connected.

If you specify 0, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

n2 specifies the number of times to retry a transmission while a station is connected.

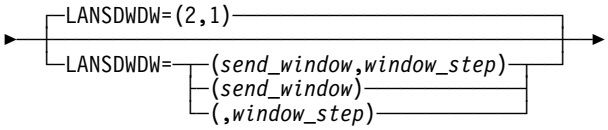
A value of 0 means that no retry will be attempted.

When the number of retries (*n2*) has been reached and the timer (*t1*) has elapsed for each retry, the link station is considered to be inoperative.

If you do not code LANRESP for this physical unit, the default values for the timer and the retry count are the values that you specify on the LANCON operand of the PORT definition statement in the LAN major node.

This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANSDWDW (VM, VSE)



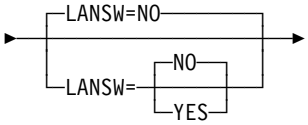
statements: PU
dependency: see description
range: 1–127 (send window)
0–send window (window step)

specifies the send window (*k*) and window step (*nw*). The send window specifies the maximum number of sequentially numbered I-LPDUs that the link station can have outstanding at any time. The window step specifies the number of sequentially numbered I-LPDUs that a link station must receive before increasing the local window.

A value of 0 for the window step (*nw*) means that the send window (*k*) is static and dynamic windowing is disabled.

This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANSW (VM, VSE)



statements: PU
dependency: see description

specifies whether the physical unit can be used as a station on a LAN.

If you do not code this operand, you must code a value for one of the following operands for the physical unit to be used as a station on the LAN:

LANACK
LANCON
LANINACT
LANRESP
LANSDWDW
MACADDR
SAPADDR.

The minimum coding required for a dial-out physical unit is MACADDR=macaddr. The minimum coding required for a dial-in physical unit is LANSW=YES.

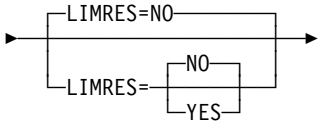
If a physical unit can be used on the LAN, additional storage is allocated to contain LAN definition values and their defaults when the physical unit is defined. This additional storage is 16 bytes for each LAN-capable physical unit that can be used on the LAN.

This operand applies only if the physical unit can be attached directly to a processor through its Token-Ring Subsystem or a processor through its multi-protocol communication subsystem.

LANSW=NO
specifies that the physical unit cannot be used as a station on the LAN.

LANSW=YES
specifies that the physical unit can be used as a station on the LAN.

LIMRES



statements: PU
dependency: LU 6.2 only

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.

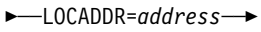
If you do not code a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended. The LIMRES operand is only applicable to PU type 1 and 2.

This operand applies to LU 6.2 conversations only.

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



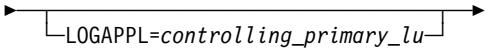
statements: LU
dependency: see note
range: 0–255 (PU Type 2)
0–63 (PU Type 1)

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



statements: LU
dependency: NetView or TSO (MVS)

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.

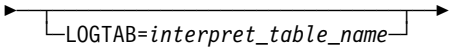
Notes:

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

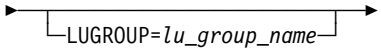
LOGTAB



statements: LU
dependency: 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 659 for a description of how to define an interpret table.

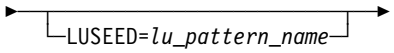
LUGROUP



statements: PU
dependency: 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



statements: PU
dependency: LUGROUP

provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

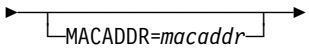
This operand is valid only when you code the LUGROUP operand.

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 “Selection of Definitions for Dependent LUs (SDDLU) Exit Routine” in *VTAM Customization*.

MACADDR (VM, VSE)



statements: PU
dependency: see description

specifies the 12-digit hexadecimal medium access control (MAC) address for the station on the ring that this PU definition statement defines. If you specify the MAC address as an 8-digit hexadecimal number, VTAM prefixes the number with 4000 to form the 12-digit address.

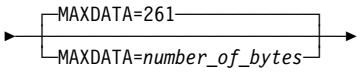
Note: The 12-digit value of MACADDR is the same as the last twelve digits of the DIALNO operand on the PATH definition statement in an NCP/Token-Ring interconnection (NTRI) definition. See page 419 for an example of the format.

X'000000000000' is not a valid MACADDR value.

This operand applies only if the physical unit can be attached directly to a processor with a local area network. MACADDR is required if the physical unit is used as a dial-out physical unit.

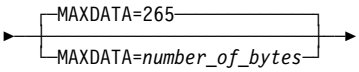
MAXDATA

For PU type 1:



statements: PU
dependency: none
range: 5-65535

For PU type 2:



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.

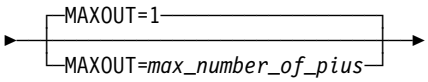
MAXDATA is ignored for type 2.1 nodes attached through an NCP with peripheral node type 2.1 support. The type 2.1 node supplies the actual value used when the connection is established.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for more information about physical units connected through an NCP.

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

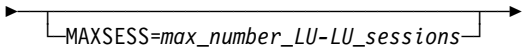


statements: PU
dependency: none
range: 1–127

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.

MAXSESS

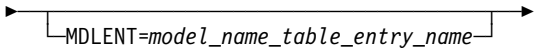


statements: GROUP, LINE, PU
dependency: independent LUs attached to an NCP
range: 1–65535

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.

MDLENT

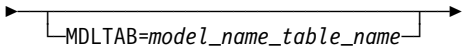


statements: LU
dependency: MDLTAB

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



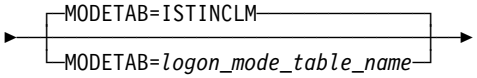
statements: LU
dependency: none

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

MODETAB



statements: LU
dependency: 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 666. 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

statements: VBUILD

statements: PU, LU

provides the name associated with the resource being defined.

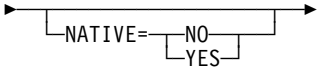
On the VBUILD: *name* is optional. If a syntax or definition error is detected with this statement during activation of the major node, *name* appears in the message issued to identify the error.

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.

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



statements: PU
dependency: BN=YES; APPN only

specifies whether this link station represents a connection to a native node.

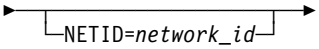
If NATIVE is not specified on either the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- 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



statements: PU
dependency: none

specifies a 1–8 character network name.

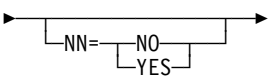
Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network identifiers. Although VTAM allows you to use #, other products might enforce this restriction.

For type 2 physical units: This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See page 600 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 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



statements: PU
dependency: APPN only

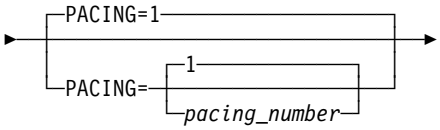
specifies whether the adjacent node is expected to be a network node.

- 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



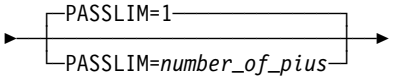
statements: LU
dependency: none
range: 0–63

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

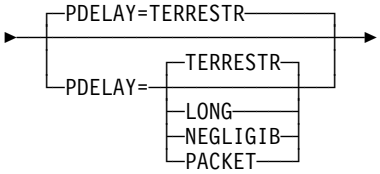
PASSLIM



statements: PU
dependency: 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.

PDELAY



statements: PU
dependency: 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.

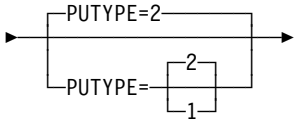
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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



statements: PU
dependency: 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.

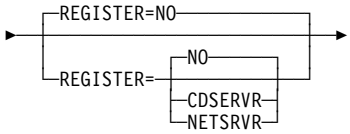
For a type 2.1 physical unit, code PUTYPE=2.

The physical unit type for a LAN switched station must be 2.

For a PU type 1, 2, or 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 431.

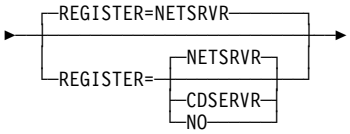
REGISTER

For Independent LUs:



statements: LU
dependency: none

For Dependent LUs:



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.

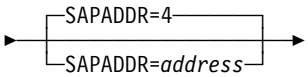
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

SAPADDR (VM, VSE)



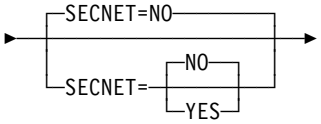
statements: PU
dependency: see description
range: 4–252

specifies the service access point (SAP) address of the physical unit that is on the ring. The hexadecimal value of SAPADDR is equivalent to the third and fourth digits of the DIALNO operand of the PATH definition statement in an NCP/Token-Ring interconnection (NTRI) definition.

Specify *n* as a decimal integer that is a multiple of 4.

This operand applies only if the physical unit can be attached directly to a processor with a local area network.

SECNET



statements: PU
dependency: 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.

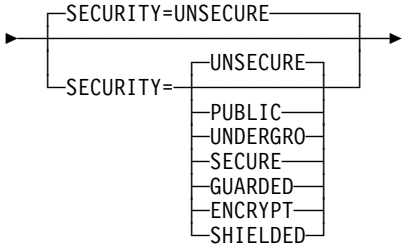
SECNET=NO
specifies that this physical unit is not associated with the secondary network.

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.

SECURITY (MVS, VM)



statements: PU
dependency: 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.

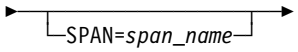
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN

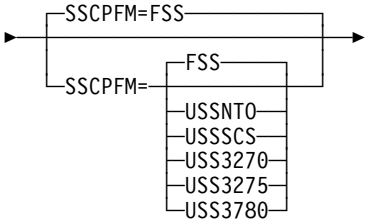


statements: PU,LU
dependency: NetView

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



statements: LU
dependency: 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.

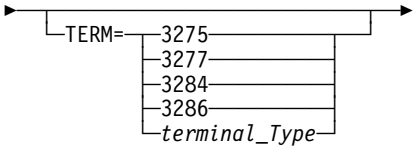
SSCPFM=USS3270
allows printing of a user-defined logon message (USSMSG10) on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

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
is 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

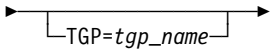


statements: LU
dependency: none

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 “Device Characteristics Field” in *VTAM Programming*.

TGP

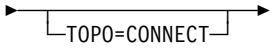


statements: PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO



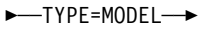
statements: PU
dependency: APPN only

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.

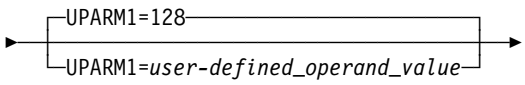
TYPE



statements: VBUILD
dependency: 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

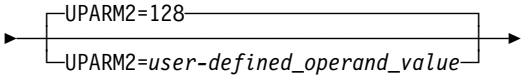


statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

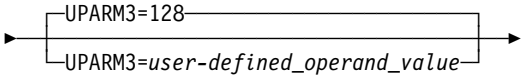
UPARM2



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

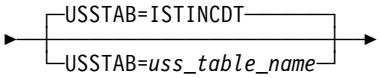
UPARM3



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

USSTAB



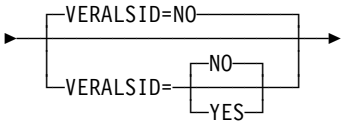
statements: LU
dependency: dependent LUs only

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 690.

USSTAB is valid for dependent logical units only.

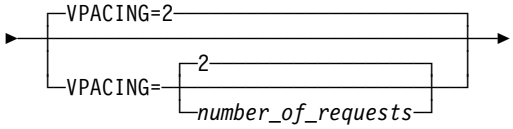
VERALSID



statements: PU
dependency: 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



statements: LU
dependency: 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 “Session Pacing” in the *VTAM Network Implementation Guide*.

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 a correct coding sequence) for the NCP major 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 347) and the NETID operand (see "NETID" on page 333) 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.

Notes on Coding the PCCU Definition Statement for VSE: More than one PCCU definition statement in the NCP definition may have the same SUBAREA value, but different NETID values. If the SUBAREA value is the same as that of the host, the first PCCU definition statement VTAM finds (without NETID specified) with that SUBAREA value is processed for that host. For this reason, be careful when providing an NCP definition to VTAM.

If more than one such host (each in a different network) with the same SUBAREA value processes the same set of NCP definitions, you must change the definitions in at least one of the hosts. Delete PCCU definition statements with matching subareas that do not apply to the host, or change their sequence so that they come after the PCCU definition statements that do apply.

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

- UNITSZ.

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 “Attaching NCP to VTAM” in the *VTAM 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.

Notes on Coding the HOST Statement for VSE: For a channel-attached NCP, the operands on the first HOST definition statement with a SUBAREA value corresponding to the host’s subarea (without NETID specified) are used. If necessary, change the operands on the HOST definition statement.

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:

- 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

"Coding the GWNAU Definition Statement" in the *VTAM 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 33 (Page 1 of 5). NCP Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PCCU	AUTODMP	302			NO	DUMPDS
		AUTOIPL	302			NO	
		AUTOSYN	303			YES	
		BACKUP	303			NO	OWNER
		CDUMPDS	306			DUMPDS value	
		CHANCON	306			COND	
		CONFGDS	307				MVS,VSE
		CONFGPW	307				MVS,VSE CONFGDS
		CUADDR	310				
		DELAY	310			0.200	
		DUMPDS	312				
		DUMPLD	313			NO	
		DUMPSTA	313				
		GWCTL	320			SHR	SNI only
		INITEST	321			NO	3705 only
		LOADFROM	325			HOST	see description
		LOADSTA	325				
		MAXDATA	330			65535	
		MDUMPDS	332			DUMPDS value	
		NCPLUB	333				VSE
		NETID	333				see description
		NEWPATH	335				
		OWNER	336				VTAM
		RNAME	343				
		SAVEMOD	344				3720 or 3745
		SUBAREA	347				
		TGN	349			1 (Subarea) ANY (APPN)	
		VFYC	353			YES	
		VFYLM	353			NO	
name	BUILD	LOADLIB	325	R			
		NETID	333	R			
		SUBAREA	347	R			VTAM & NCP
		COSTAB	308			ISTVTCOS	
		MAXSUBA	331				Required if PRE-ENA node in network VTAM & NCP
		NEWNAME	335				
		PUNAME	340				
		SALIMIT	343			255	
name	SYSCNTRL	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					
name	HOST	INBFRS	321	R			
		MAXBFRU	329	R			
		UNITSZ	350	R			*bufsize* value on IOBUF
		NETID	333				
		SUBAREA	347			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 33 (Page 2 of 5). NCP Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	NETWORK	NETID	333	R			SNI only; COPIES
		SUBAREA	347	R			SNI only
		COPIES	308				SNI only; NETID
		COSTAB	308			ISTVTCOS	SNI only
		MAXSUBA	331				For pre-ENA; SNI only; VTAM & NCP
		SALIMIT	343			255	SNI only
name	FRSESET	SUBPORTS	348	R			VTAM & NCP; PU type 1
name	GWNAU	See the *NCP, SSP, and EP Resource Definition Reference* for operand details.					
name	GROUP	LNCTL	324	R			VTAM & NCP
		DIAL	311			NO	VTAM & NCP
		DYNPU	315			NO	
		DYNPUPFX	315			CN	DYNPU=YES
		ECLTYPE	316				VTAM & NCP
		FRELAY	318				VTAM & NCP; PU type 1
		PHYSRSC	339				
		PUTYPE	341			5	VTAM & NCP
		SHM	346			NO	VTAM
		X21SW	356			NO	VTAM & NCP VSE
name	LINE	ANSWER	300		G	ON	VTAM
		AUTO	301		G		VTAM & NCP
		AUTODL	301		G	YES	VTAM & NCP
		BUFCAP	304		G	1	VTAM
		CALL	305		G	IN	VTAM & NCP
		ISTATUS	322		G	ACTIVE	VTAM
		LINEAUT	323		G		VTAM & NCP
		LOCADD	326				VTAM & NCP
		POLIMIT	339		G	(1,NOWAIT)	VTAM & NCP
		PU	340		G	YES	VTAM
		SWNORDER	348		G	CPNAME	VTAM
		TERM	348		G		VTAM & NCP
		USE	351		G		VTAM & NCP USE=REDEF USE=SPARE
		VNGROUP	354		G		VNNAME; VTAM only ECLTYPE=PHYSICAL
		VNNAME	354		G		VNGROUP; VTAM only ECLTYPE=PHYSICAL
name	PU	ADDR	299				VTAM & NCP
		ANS	299		G,L	STOP	VTAM & NCP
		AUTHLEN	301		G,L	YES	PU Type 2.1
		CAPACITY	305		G,L	8K	APPN only
		CONNTYPE	307		G,L	CONNTYPE start option	APPN only

Table 33 (Page 3 of 5). NCP Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		COSTBYTE	309		G,L	0	APPN only
		COSTTIME	309		G,L	0	APPN only
		CPCP	309		G,L	CPCP start option	APPN only
		CPNAME	309				
		DATMODE	310		G,L	HALF	VTAM & NCP
		DISCNT	311		G,L	(NO)	VTAM
		DYNADJCP	314		G,L	DYNADJCP start option	APPN only CPNAME
		DYNLU	314		G,L	DYNLU start option	PUTYPE=1, 2; VTAM
		GP3174	320		G,L	NO	VTAM & NCP
		INNPORT	321			NO	VTAM
		ISTATUS	322		G	ACTIVE	VTAM
		LIMRES	323		G,L	NO	VTAM; LU 6.2
		LMODADR	324			see description	VTAM & NCP LPDATS; SEGMENT
		LPDA	328			ALLOW	VTAM & NCP
		LUGROUP	329				VTAM
		LUSEED	329				VTAM
		NATIVE	333		G,L		BN=YES APPN only
		NETID	333				VTAM & NCP
		NN	336		G,L		APPN only
		OWNER	336		G,L		VTAM only; PUTYPE=1 or 2
		PDELAY	338		G,L	TERRESTR	APPN only
		PUDR	340		G,L		VTAM & NCP
		PUTYPE	341		G,L	5	VTAM & NCP
		RETRIES	342		G,L		VTAM & NCP
		SECNET	344		L	NO	VTAM
		SECURITY	345		G,L	UNSECURE	APPN only MVS,VM
		SEGMENT	345				VTAM & NCP
		TGN	349				VTAM & NCP
		TGP	350		G,L		APPN only
		TOPO	350			CONNECT	APPN only
		UPARM1	351		G,L	128	APPN only
		UPARM2	351		G,L	128	APPN only
		UPARM3	351		G,L	128	APPN only
		VERALSID	352		G,L	NO	APPN only
		XID	355		G,L	NO	VTAM & NCP
name	LU	LOCADDR	326	R			VTAM & NCP
		ASLENT	300		G,L,P		ASLTAB; VTAM
		ASLTAB	300		G,L,P		VTAM
		CLRSESSQ	306		G,L,P	NO	VTAM
		DLOGMOD	312		G,L,P		VTAM
		ENCR	316		G,L,P	NONE	VTAM MVS
		FASTPASS	317		G,L,P	YES	VTAM; TSO

Table 33 (Page 4 of 5). NCP Major Node Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		FEATUR2	317		G,L,P	(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	VTAM
		ISTATUS	322		G	ACTIVE	VTAM
		LOGAPPL	327		G,L,P		VTAM
		LOGTAB	328		G,L,P		VTAM; dependent LUs
		LUDR	328		G,L,P		VTAM & NCP
		MAXSESS	330		G,L,P		VTAM & NCP Independent LUs only
		MDLENT	331		G,L,P		VTAM; MDLTAB
		MDLTAB	331		G,L,P		VTAM
		MODETAB	332		G,L,P	ISTINCLM	VTAM
		PACING	337		G,L,P	(1,1,ADAP)	VTAM & NCP
		REGISTER	341		P	see description	APPN only
		RESSCB	342			0	VTAM & NCP
		SSCPFM	346		G,L,P	FSS	VTAM
		TERM	348		P		
		USSTAB	352		G,L,P	ISTINCDT	VTAM; dependent LUs
		VPACING	354		G,L,P	(2,1,ADAP)	VTAM
name	CLUSTER	GPOLL	320				VTAM & NCP
		ISTATUS	322		G	ACTIVE	VTAM
name	TERMINAL	TERM	348	R	G,L,C		VTAM & NCP
		ASLENT	300		G,L,C		ASLTAB
		ASLTAB	300		G,L,C		
		BHSET	304		G,L,C		VTAM & NCP
		DLOGMOD	312		G,L,C		VTAM; MODETAB
		FEATUR2	317		G,L,C	(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	VTAM
		ISTATUS	322		G	ACTIVE	VTAM
		LOGAPPL	327		G,L,C		VTAM; NetView or TSO (MVS)
		LOGTAB	328		G,L,C		dependent LUs
		MDLENT	331		G,L,C		MDLTAB
		MDLTAB	331		G,L,C		
		MODETAB	332		G,L,C	ISTINCLM	
		SSCPFM	346		G,L,C	FSS	
		USSTAB	352		G,L,C	ISTINCDT	dependent LUs

Table 33 (Page 5 of 5). NCP Major Node Statements and Operands

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.

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 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	LU
CAPACITY	ASLENT
COSTBYTE	ASLTAB
COSTIME	CLRSESSQ
CPCP	DLOGMOD
DISCNT	ENCR
DYNADJCP	FASTPASS
DYNLU	FEATUR2
INNPORT	ISTATUS
ISTATUS	LOGAPPL
LIMRES	LOGTAB
LUGROUP	MDLENT
LUSEED	MDLTAB
MAXSESS	MODETAB
NN	REGISTER
PDELAY	SSCPFM
SECNET	TERM
SECURITY	USSTAB
TGP	VPACING
TOPO	
UPARM1	
UPARM2	
UPARM3	
VERALSID	
VPACING	

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.

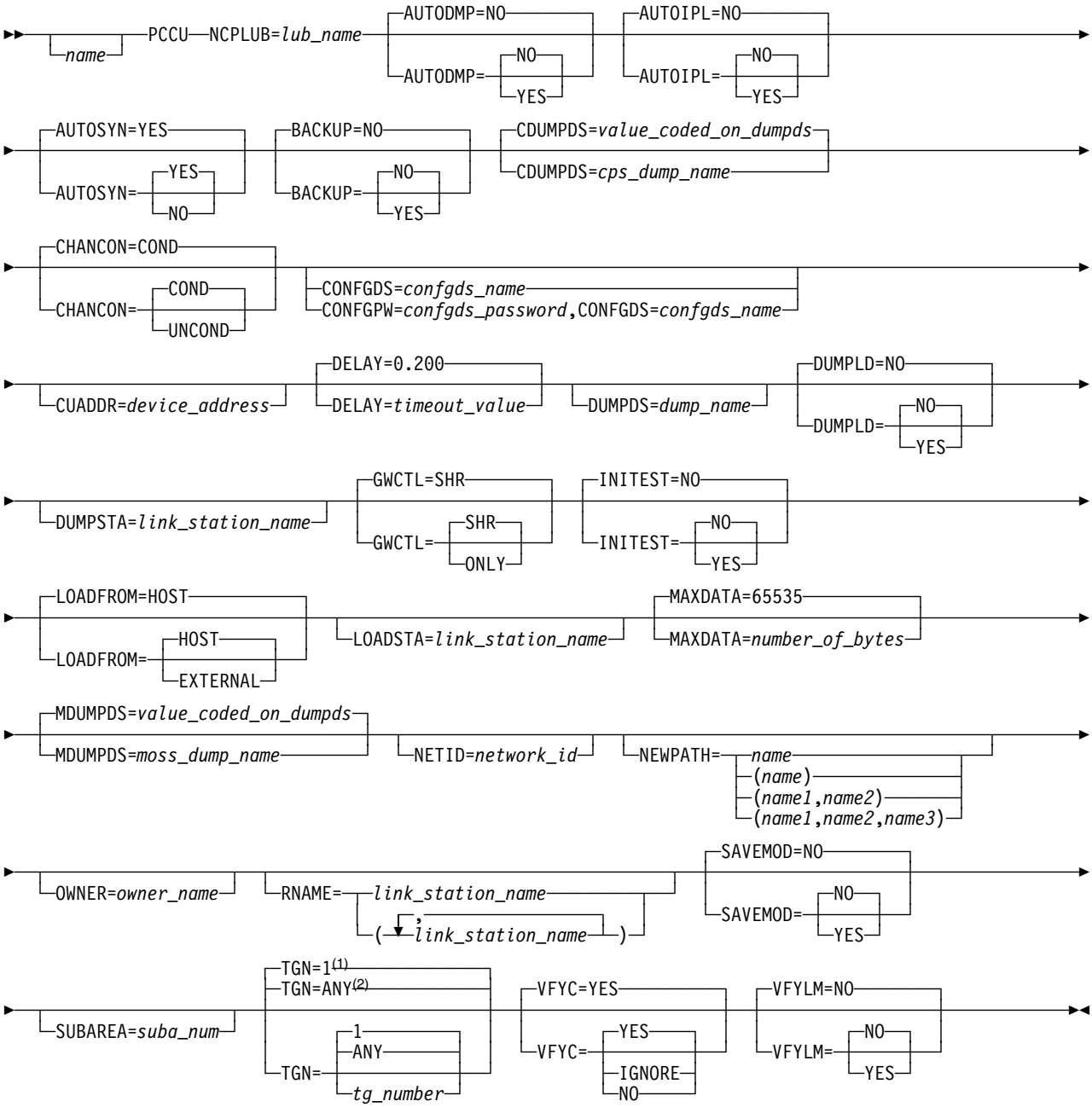
Refer to “Dynamic Reconfiguration and Change of Operands” in the *VTAM Network Implementation Guide* and “VARY ACT Command” in *VTAM Operation* for more information on the VARY ACT,UPDATE=ALL command.

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

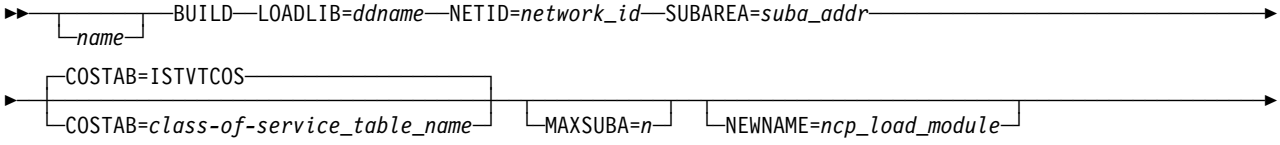


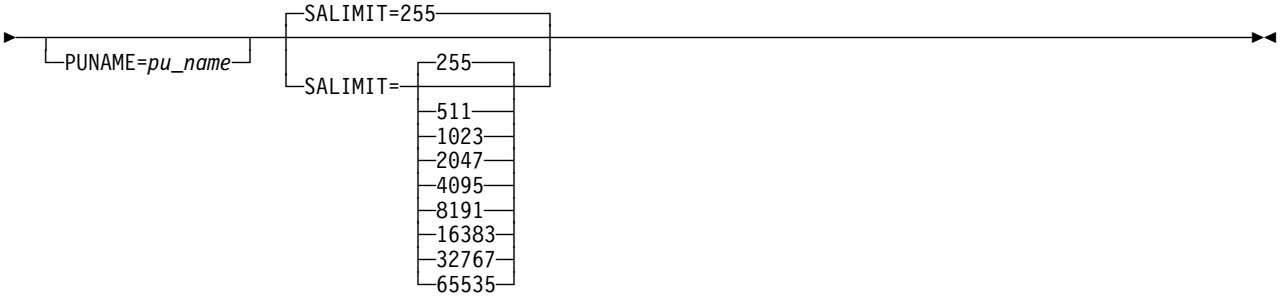
Notes:

1 TGN defaults to 1 for subarea.

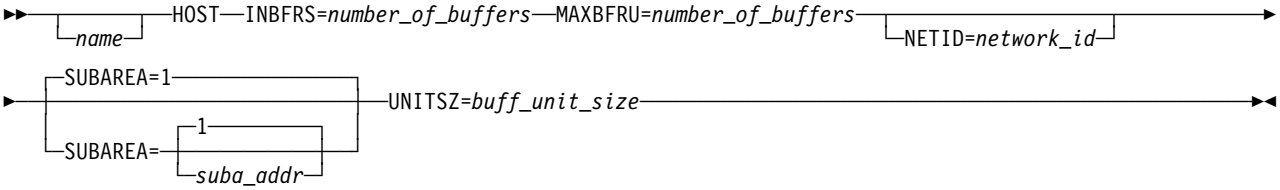
2 TGN defaults to ANY for APPN.

BUILD

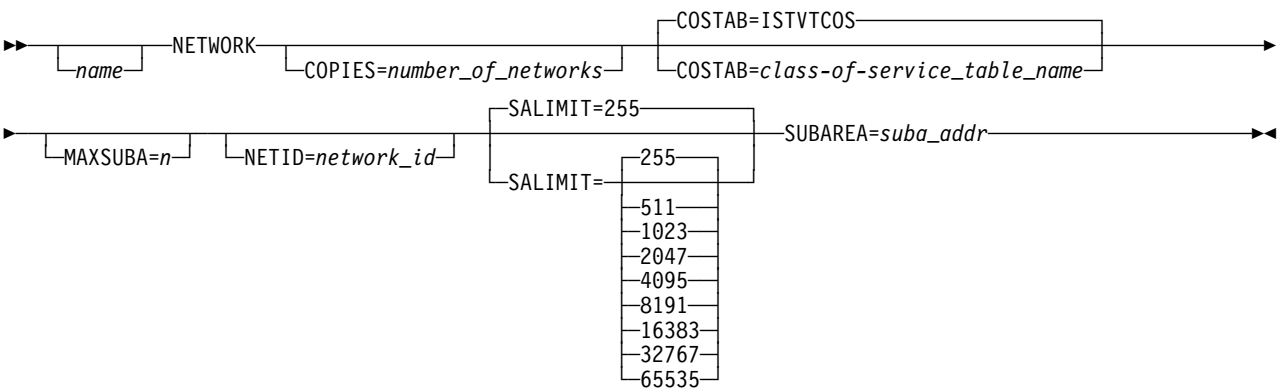




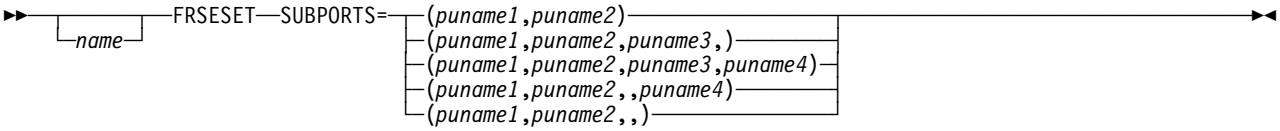
HOST



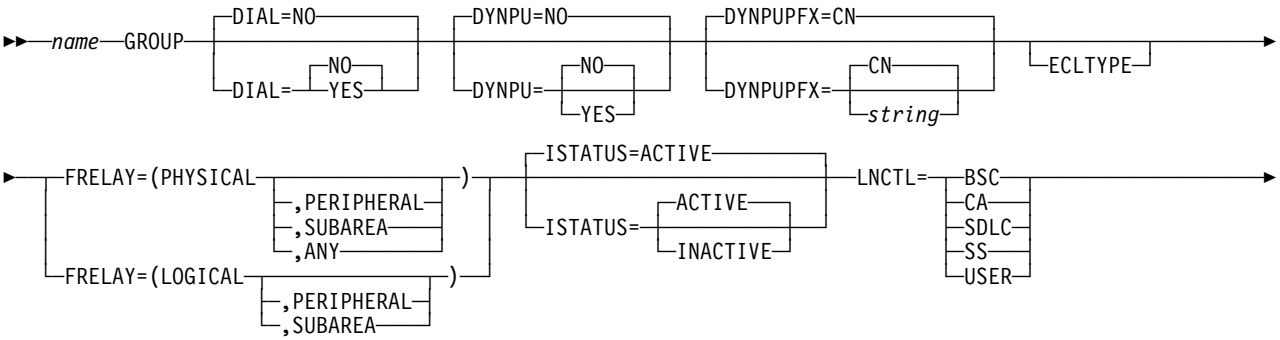
NETWORK

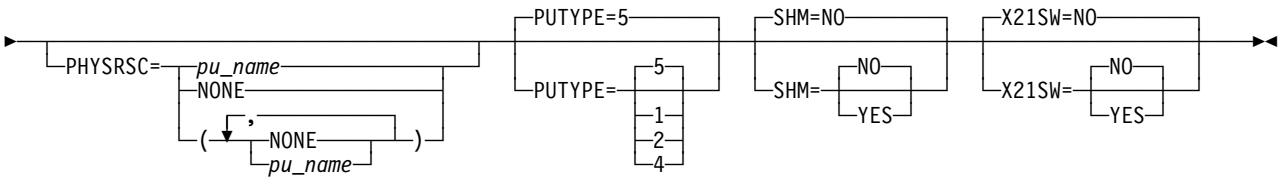


FRSESET

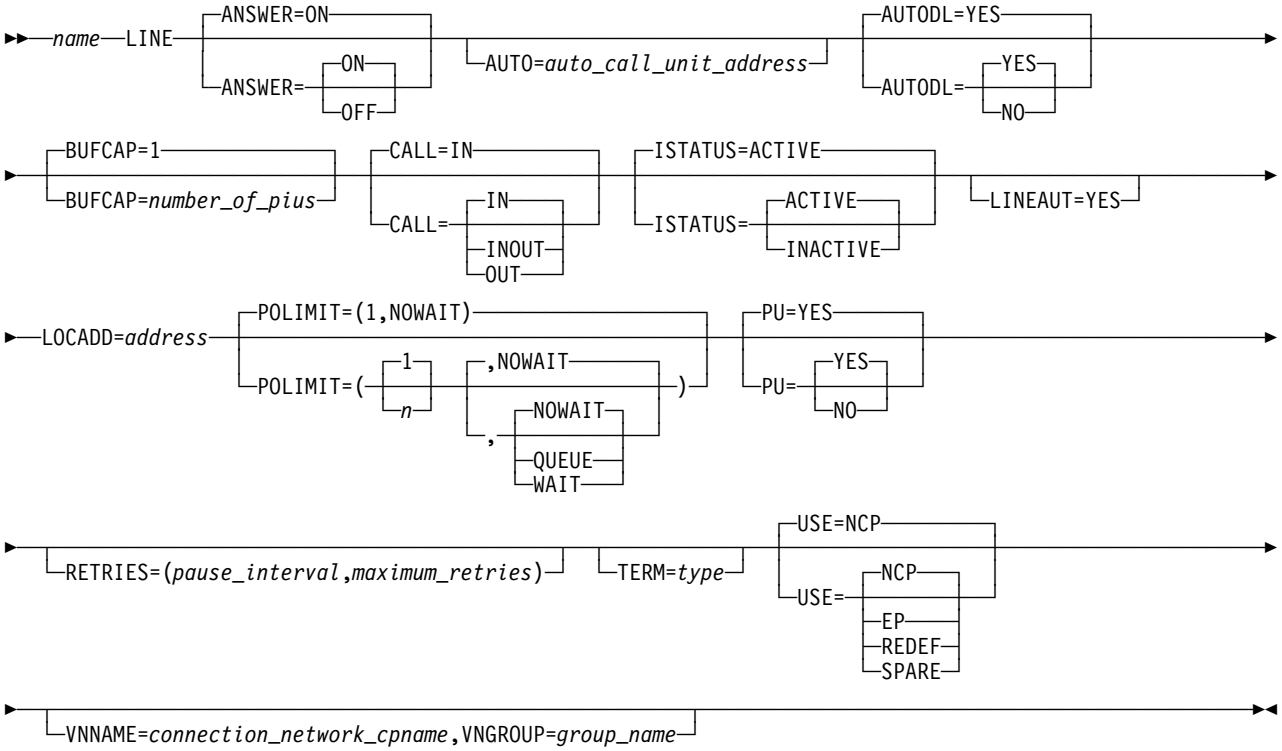


GROUP

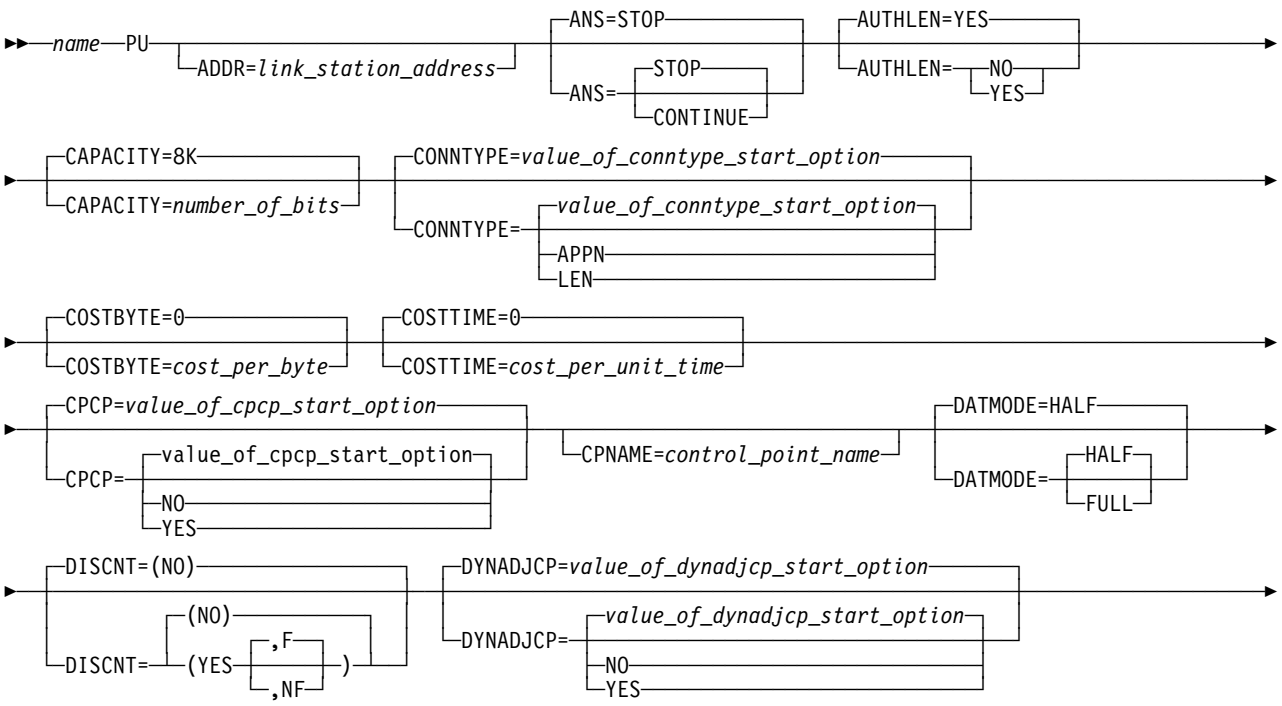


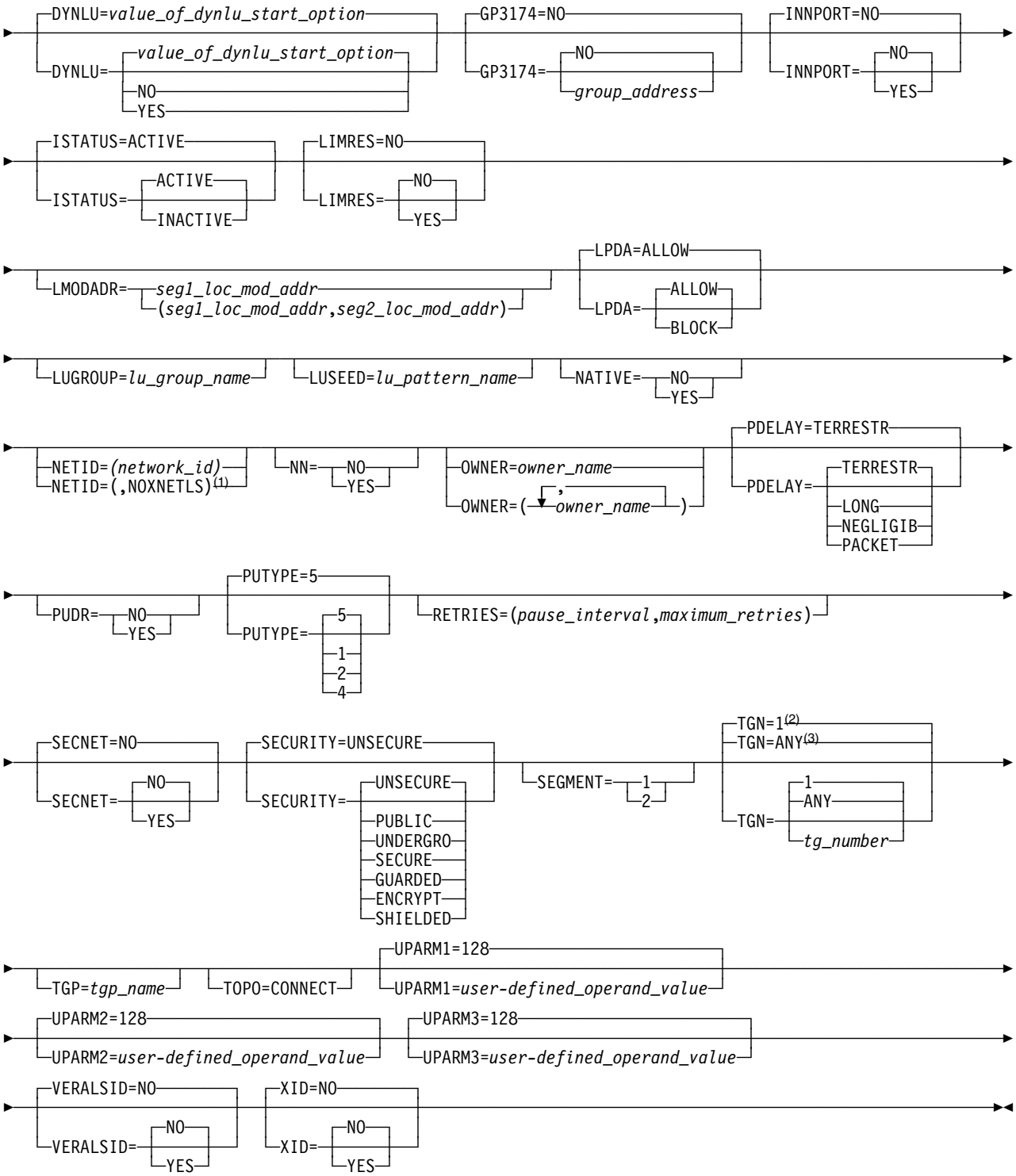

PHYSRSC=
pu_name
NONE
(
,
NONE
pu_name
)
PUTYPE=5
PUTYPE=
5
1
2
4
SHM=NO
SHM=
NO
YES
X21SW=NO
X21SW=
NO
YES

LINE


name
LINE
ANSWER=ON
ANSWER=
ON
OFF
AUTO=auto_call_unit_address
AUTODL=YES
AUTODL=
YES
NO
BUFCAP=1
BUFCAP=number_of_pius
CALL=IN
CALL=
IN
INOUT
OUT
ISTATUS=ACTIVE
ISTATUS=
ACTIVE
INACTIVE
LINEAUT=YES
LOCADD=address
POLIMIT=(1,NOWAIT)
POLIMIT=(
1
n
,NOWAIT
,
NOWAIT
QUEUE
WAIT
)
PU=YES
PU=
YES
NO
RETRIES=(pause_interval,maximum_retries)
TERM=type
USE=NCP
USE=
NCP
EP
REDEF
SPARE
VNNAME=connection_network_cpname,VNGROUP=group_name

PU


name
PU
ADDR=link_station_address
ANS=STOP
ANS=
STOP
CONTINUE
AUTHLEN=YES
AUTHLEN=
NO
YES
CAPACITY=8K
CAPACITY=number_of_bits
CONNTYPE=value_of_conntype_start_option
CONNTYPE=
value_of_conntype_start_option
APPN
LEN
COSTBYTE=0
COSTBYTE=cost_per_byte
COSTTIME=0
COSTTIME=cost_per_unit_time
CPCP=value_of_cpcp_start_option
CPCP=
value_of_cpcp_start_option
NO
YES
CPNAME=control_point_name
DATMODE=HALF
DATMODE=
HALF
FULL
DISCNT=(NO)
DISCNT=
(NO)
(YES
,F
,NF
)
DYNADJCP=value_of_dynadjcp_start_option
DYNADJCP=
value_of_dynadjcp_start_option
NO
YES



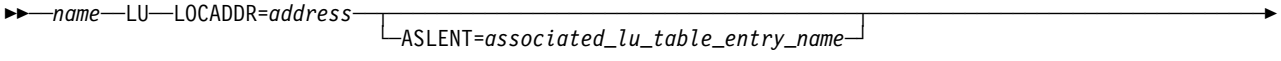
Notes:

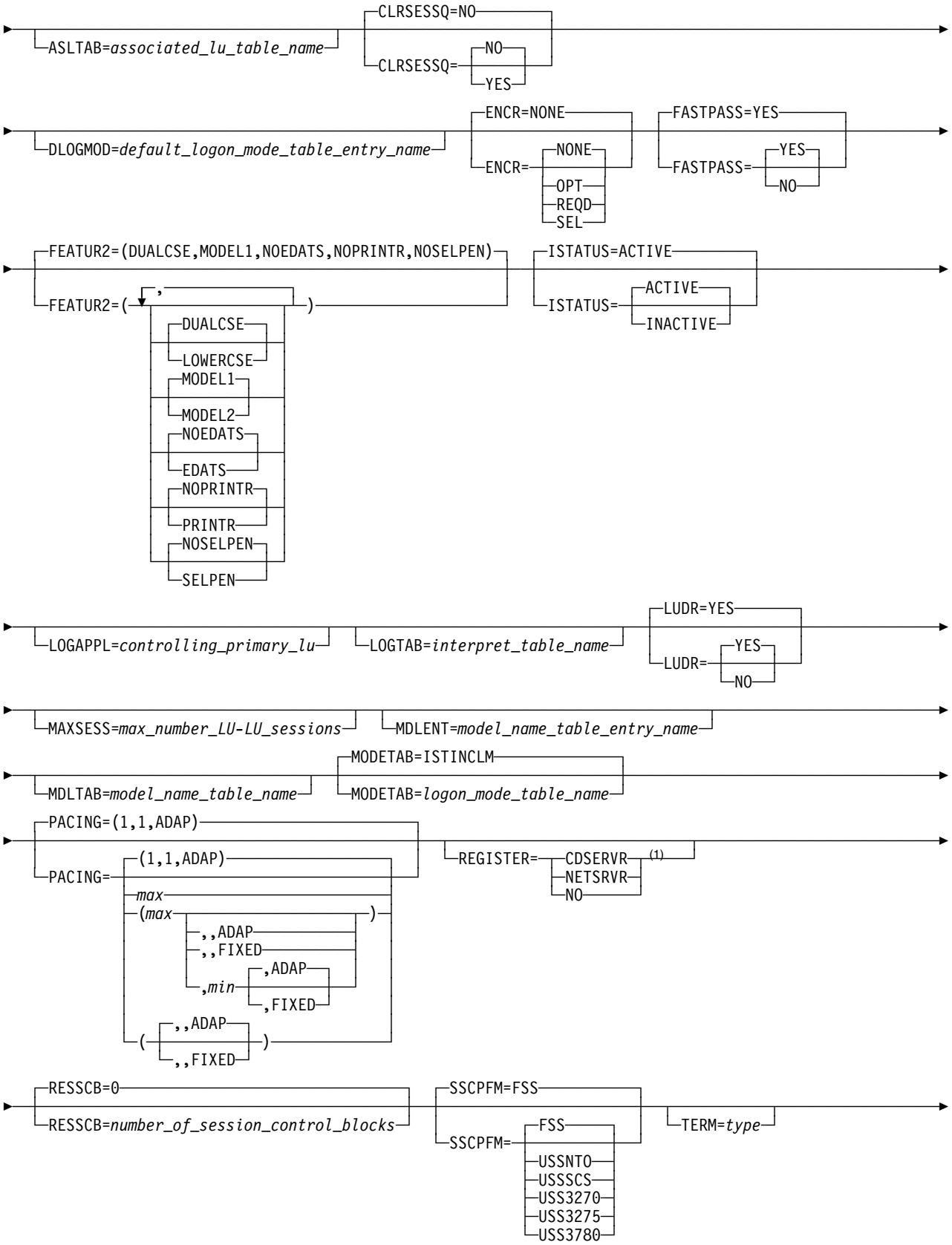
1 NOXNETLS applies only to the PU definition statement.

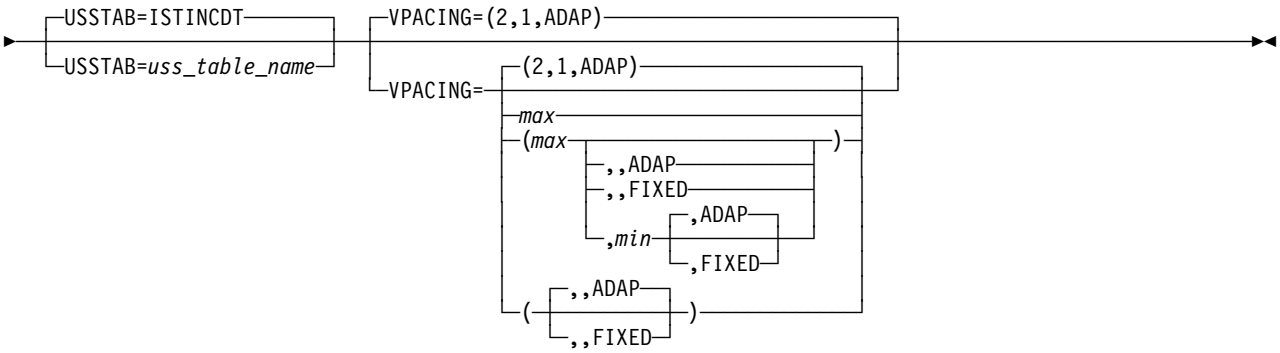
2 TGN defaults to 1 for subarea.

3 TGN defaults to ANY for APPN.

LU

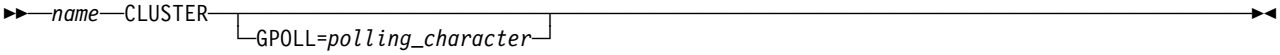



ASLTAB=associated_lu_table_name
CLRSESSQ=NO
CLRSESSQ=
NO
YES
DLOGMOD=default_logon_mode_table_entry_name
ENCR=NONE
ENCR=
NONE
OPT
REQD
SEL
FASTPASS=YES
FASTPASS=
YES
NO
FEATUR2=(DUALCSE,MODEL1,NOEDATS,NOPRINTR,NOSELPEN)
FEATUR2=(
,
)
DUALCSE
LOWERCSE
MODEL1
MODEL2
NOEDATS
EDATS
NOPRINTR
PRINTR
NOSELPEN
SELPEN
ISTATUS=ACTIVE
ISTATUS=
ACTIVE
INACTIVE
LOGAPPL=controlling_primary_lu
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
PACING=(1,1,ADAP)
PACING=
(1,1,ADAP)
max
(max
,,ADAP
,,FIXED
,min
,ADAP
,FIXED
)
(
,,ADAP
,,FIXED
)
REGISTER=
CDSERVR
NETSRVR
NO
(1)
RESSCB=0
RESSCB=number_of_session_control_blocks
SSCPFM=FSS
SSCPFM=
FSS
USSNTO
USSSCS
USS3270
USS3275
USS3780
TERM=type

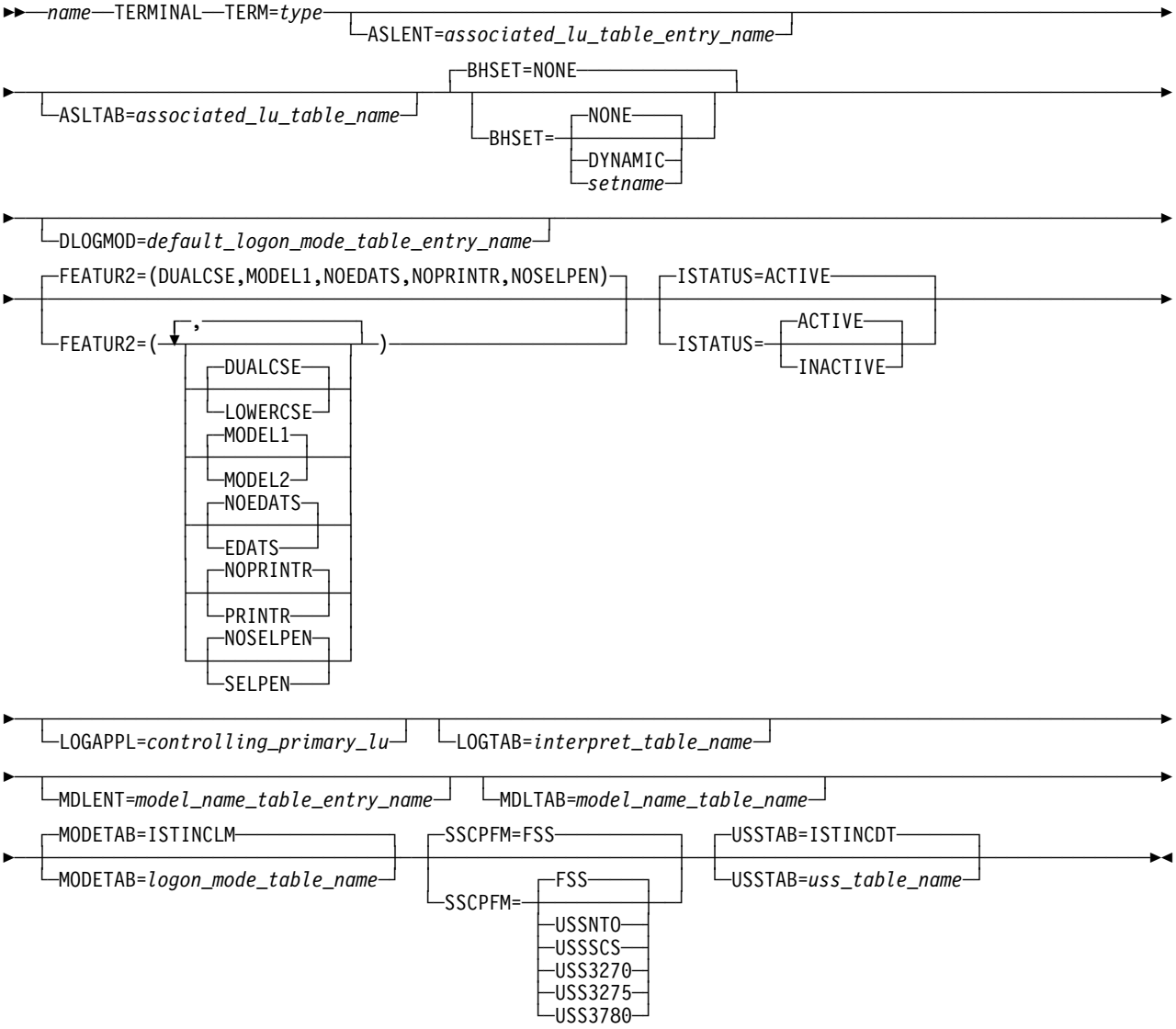


Note:
[1] The default for REGISTER depends on whether you code it for a dependent or independent LU. See operand description.

CLUSTER



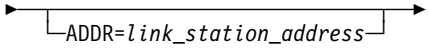
TERMINAL



Operand Descriptions

The following section describes the operands you can code for the network control program (NCP) major node.

ADDR



statements: PU
dependency: VTAM and NCP

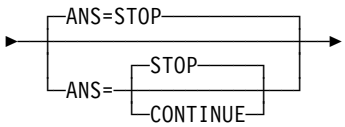
specifies the 2-character hexadecimal representation of the 8-bit address of a PU on a nonswitched line. Any bit configuration except X'00' or X'FF' is valid.

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.

ANS



statements: GROUP, LINE, PU
dependency: 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

- MVS If the terminal participates in XRF sessions
- For frame relay PUs.

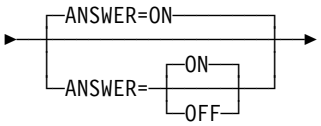
When a session is continued and takeover occurs, if the LU taken over is a dependent logical unit, it must support ACTLU(ERP), or the session is terminated during takeover processing. An independent logical unit does not receive ACTLU, so this restriction does not apply.

Note that there are several restrictions for session continuity in certain configurations. For information on these restrictions, refer to “SSCP Takeover” in the *VTAM 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



statements: GROUP, LINE
dependency: VTAM only

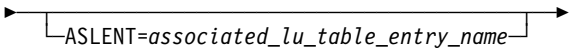
specifies whether physical units can dial in to the NCP. ANSWER is valid only if the SDLC line has dial-in capability.

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 "VARY ANS Command" in *VTAM Operation* for information on using the VARY ANS command.

ASLENT

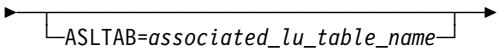


statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; ASLTAB

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



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only

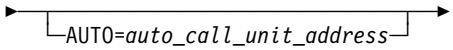
specifies the name of the associated LU table to be used for the logical unit. See "Associated LU Table" on page 638 for a description of how to define an associated LU table.

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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

AUTO

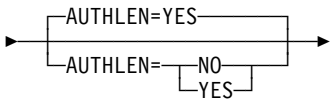


statements: LINE
dependency: VTAM and NCP

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



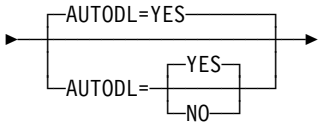
statements: GROUP, LINE, PU
dependency: PU Type 2.1

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



statements: GROUP, LINE
dependency: VTAM and NCP

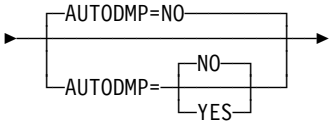
specifies whether CCITT X.21 data circuit-terminating equipment (DCE) accepts dial digits.

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



statements: PCCU
dependency: 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.

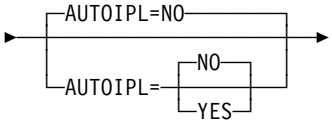
The AUTODMP operand is valid only if you code a DUMPDS operand in this PCCU definition statement. Otherwise, no NCP dump can be taken. (See “DUMPDS” on page 312 for a description of DUMPDS.) After the dump, (or dumps, if more than one host takes a dump), VTAM interprets the AUTOIPL setting to determine 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.

MVS 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



statements: PCCU
dependency: 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.

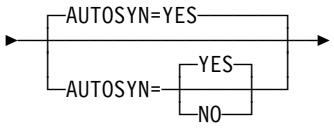
AUTOIPL=YES
indicates that another copy of the same NCP is automatically loaded and restarted.

MVS 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



statements: PCCU
dependency: none

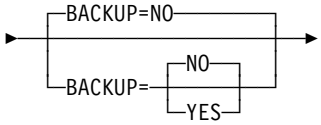
used to determine whether the NCP load module specified replaces the current load module in the communication controller. This operand applies only if the VTAM operator requests (or uses by default) LOAD=U on the VARY ACT command for an NCP. VTAM uses this operand when the specified load module name 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 353 for a description of VFYLM.)

BACKUP



statements: PCCU
dependency: OWNER

determines whether this NCP's resources can be taken over by a backup host if the owning host fails.

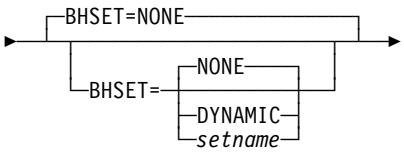
BACKUP=NO
indicates that the backup host cannot take over the NCP's resources.

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 336 for a description of OWNER.)

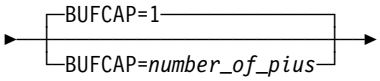
BHSET



statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: 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



statements: GROUP, LINE
dependency: VTAM only
range: 1–255

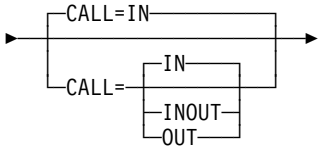
specifies the maximum number of PIUs that can be read at one time from a line without retransmissions, assuming each PIU is the maximum length allowed (MAXBFRU(MAX) × IOBUFSIZE). If an individual PIU length is smaller than the maximum size allowed for a PIU, more PIUs than BUFCAP's value could be read in. The greatest number of buffers that VTAM will allocates 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 tone 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



statements: GROUP, LINE
dependency: 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.

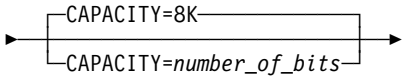
CALL=INOUT
specifies that the line is used for connections initiated by terminals, in addition to connections initiated by VTAM, the VTAM operator, or application programs.

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

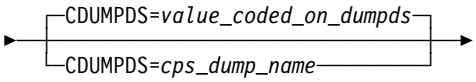


statements: GROUP, LINE, PU
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CDUMPDS



statements: PCCU
dependency: none

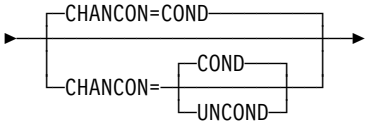
names the file that contains communication scanner processor (CSP) dumps.

MVS *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. **VM** *cps_dump_name* must be on a FILEDEF statement. **VSE** Use the device name (SYSxxx) that is coded on the EXTENT or ASSIGN statement for the DLBL statement that contains the file name for the dump dataset.

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



statements: PCCU
dependency: none

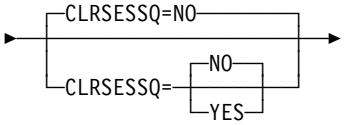
determines whether VTAM sends a conditional or an unconditional channel contact request to the NCP when the host is to activate the NCP over a channel.

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 78 for a description of the CHANCON operand in a channel-attachment major node.

CLRSESSQ



statements: GROUP, LINE, PU, LU
dependency: VTAM only

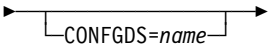
specifies whether a dependent LU is forced into an inhibited session state when the LU is disabled.

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.

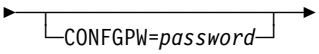
CONFGDS (MVS, VSE)



statements: PCCU
dependency: 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 (MVS, VSE)

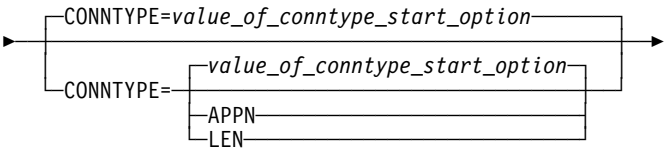


statements: PCCU
dependency: CONFGDS

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



statements: GROUP, LINE, PU
dependency: APPN only

specifies for a PU 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.

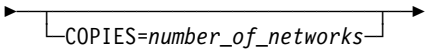
When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

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.

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.

COPIES

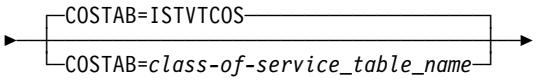


statements: NETWORK
dependency: SNI only; NETID

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



statements: BUILD, NETWORK
dependency: none

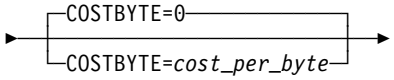
names a class-of-service (COS) table for the network identified by NETID. If you omit this operand, the default COS table is ISTSDCOS. This operand is processed regardless of which network corresponds to the NETWORK definition statement.

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
```

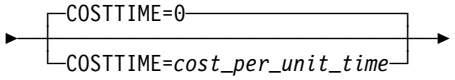
COSTBYTE



statements: GROUP, LINE, PU
dependency: 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.

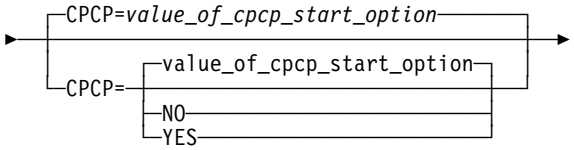
COSTTIME



statements: GROUP, LINE, PU
dependency: 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



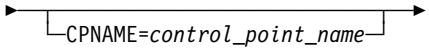
statements: GROUP, LINE, PU
dependency: APPN only

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.

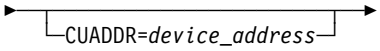
CPNAME



statements: PU
dependency: 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



statements: PCCU
dependency: none

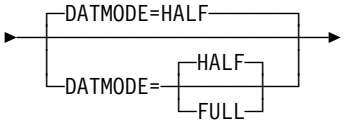
defines the 3-digit or 4-digit (MVS) hexadecimal channel unit address of the channel attachment for the communication controller in which the NCP runs.

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 74 for a description of the ADDRESS operand in a channel-attachment major node.

DATMODE



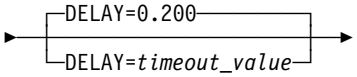
statements: GROUP, LINE, PU
dependency: VTAM and NCP

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



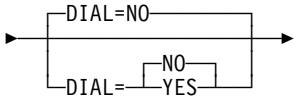
statements: PCCU
dependency: none
range: 0-9.999

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 “DELAY Operand” in the *VTAM 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 82 for a description of the DELAY operand in a channel-attachment major node.

DIAL



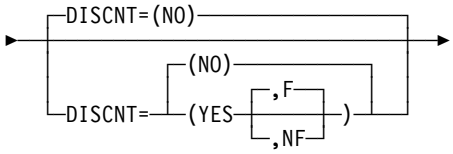
statements: GROUP
dependency: VTAM and NCP

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



statements: GROUP, LINE, PU, CLUSTER
dependency: 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 “Switched SDLC Connection” in the *VTAM Network Implementation Guide* for a discussion of those conditions.

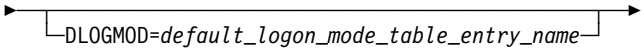
DISCNT=(YES)
If you code YES, VTAM automatically ends the SSCP-LU and SSCP-PU sessions when the last LU-LU session is ended.

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.

DLOGMOD

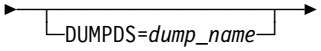


statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; MODETAB

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 666.

DUMPDS



statements: PCCU
dependency: 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.

MVS *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. **VM** *dump_name* must be on a FILEDEF statement. **VSE** Use the device name (SYSxxx) that is coded on the EXTENT or ASSIGN statement for the DLBL statement that contains the file name for the dump dataset.

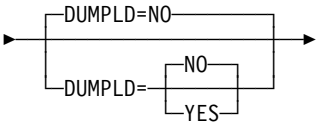
Code DUMPDS if any of the following are true:

- 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 "DISPLAY NCPSTOR Command" in *VTAM Operation*.

DUMPLD



statements: PCCU
dependency: 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.

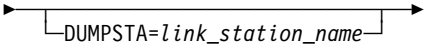
DUMPLD=YES
specifies that the NCP that has abended is automatically dumped and then reloaded from that disk.

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 "VARY ACT Command" in *VTAM Operation* for additional information on the VARY ACT command.

DUMPSTA



statements: PCCU
dependency: 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 "Channel-Attached NCP" in the *VTAM*

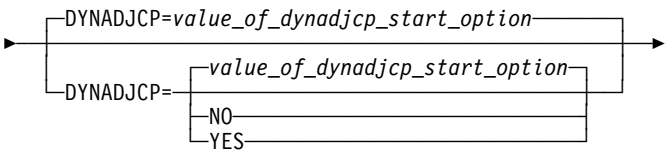
Network Implementation Guide for information on how VTAM chooses a default link station.

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 channel unit address and adding "-S." If the channel unit address is 3-digits, a leading zero is added to the name. For example, the channel unit address 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 "VARY ACT Command" in *VTAM Operation*.

DYNADJCP



statements: GROUP, LINE, PU
dependency: APPN only; ignored if CPNAME coded.

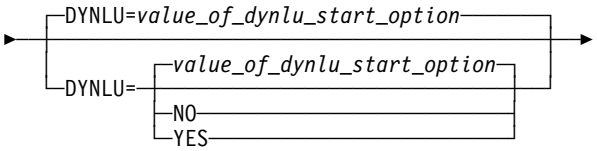
specifies whether an adjacent control point can be created dynamically for the adjacent node.

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



statements: GROUP, LINE, PU
dependency: 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,

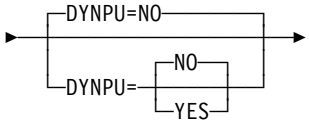
an independent logical unit that was dynamically defined using CDRSC definitions over another adjacent link station cannot use that particular PU to connect to a session.

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



statements: GROUP
dependency: 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.

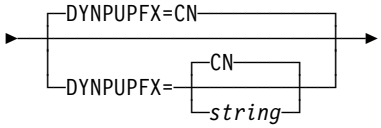
A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:

- 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



statements: GROUP
dependency: 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 10.

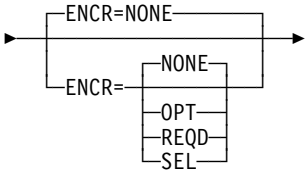
ECLTYPE

statements: GROUP
dependency: 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.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the ECLTYPE operand.

ENCR (MVS)



statements: GROUP, LINE, PU, LU
dependency: VTAM only

specifies whether this logical unit has any special requirements for enciphering and deciphering messages.

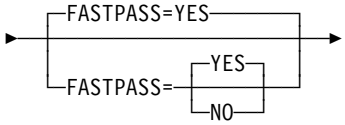
ENCR=NONE
specifies that this logical unit cannot engage in cryptographic sessions.

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.

FASTPASS (MVS)



statements: GROUP, LINE, PU, LU
dependency: 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.

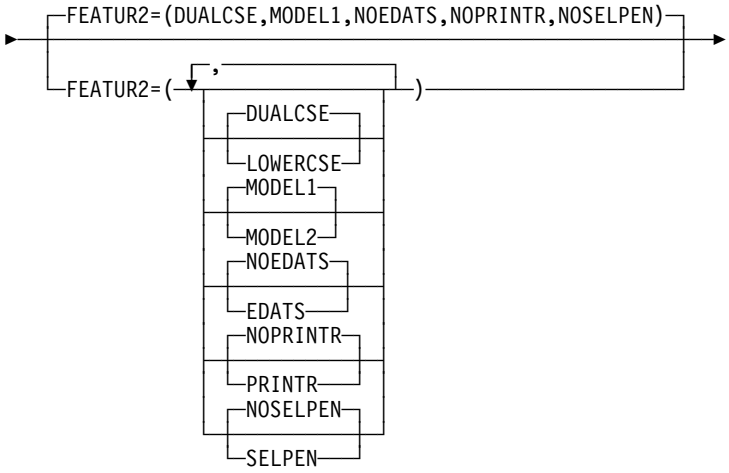
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: 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.

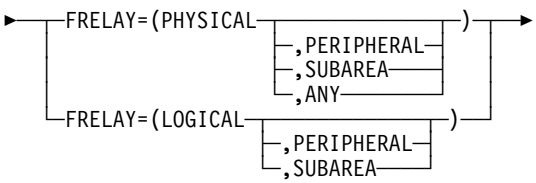
FEATUR2=MODEL1
FEATUR2=MODEL2
identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

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 “INQUIRE-Obtain Logical Unit Information or Application Program Status” in *VTAM 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



statements: GROUP
dependency: 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 PU under a frame-relay line, with the exception of the first PU (the local maintenance interface or LMI), 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.

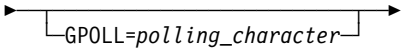
If you code FRELAY=PHYSICAL and do not specify SUBAREA, PERIPHERAL, or ANY, the default for the second suboperand depends on the line address specified by ADDRESS on the LINE definition statement. Refer to the *NCP, SSP, and EP Resource Definition Reference* for information about the ADDRESS operand.

- If the line address is 1–895 then the default is (PHYSICAL,SUBAREA).
- If the line address is 1024–1039 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 the PHYSRSC=NONE is not coded on the PU definition statement, the PU definition statement must be included in a line group with PHYSICAL specified in the FRELAY operand.

GPOLL

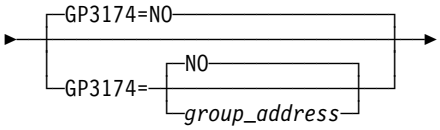


statements: CLUSTER
dependency: VTAM and NCP

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



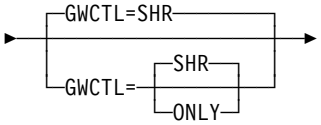
statements: GROUP, LINE, PU
dependency: VTAM and NCP

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'. If you code LPDATS=LPDA2, X'FD' is not valid.

You can change the value of GP3174 for SDLC nonswitched PUs by dynamically adding the PUs using the DR ADD definition statement.

GWCTL (MVS, VM)



statements: PCCU
dependency: 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.

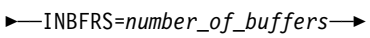
If the session setup path crosses three networks within a single gateway NCP, and if one or more gateway SSCPs are in session with the gateway NCP, the gateway SSCP in the center network must be designated to perform all of the gateway functions. Code GWCTL=ONLY on the PCCU statement for the gateway 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 “Handling Class-of-Service Tables” in the *VTAM Network Implementation Guide*.

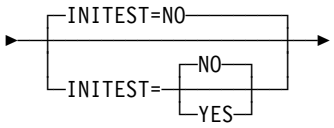
INBFRS



statements: HOST
dependency: none
range: 1–255

defines the number of buffers to be allocated by an NCP to receive data from a host.

INITEST



statements: PCCU
dependency: 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.

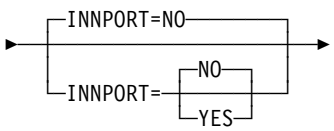
INITEST=NO
specifies that VTAM will not load the initial test routine.

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



statements: PU
dependency: 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.

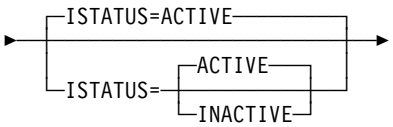
For frame-relay PUs, code INNPORT=YES.

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



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: 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).

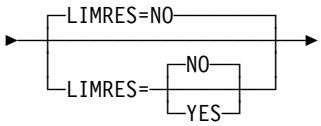
The operator can override the ISTATUS specification using the SCOPE operand on the VARY ACT command.

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



statements: GROUP, LINE, PU
dependency: VTAM only; LU 6.2 only

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.

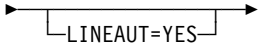
If you do not code a value for LIMRES, resources are not considered to be limited resources. Furthermore, sessions will remain active even after all conversations have ended.

This operand applies to LU 6.2 conversations only.

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

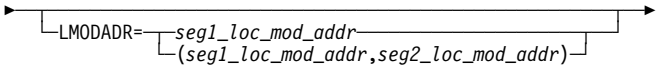


statements: GROUP, LINE
dependency: VTAM and NCP

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.

LMODADR



statements: PU
dependency: VTAM and NCP;
LPDATS; SEGMENT
range: X'0'–X'F'

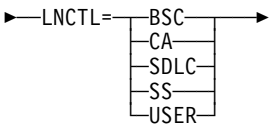
specifies the local modem addresses used on the line going to the physical unit. LMODADR is valid on the PU definition statement only if you specify LPDATS=LPDA2 on the LINE definition statement.

Specify a local modem address (*seg1_loc_mod_addr*) for link segment 1 and, if necessary, a local modem address (*seg2_loc_mod_addr*) for link segment 2. If SEGMENT=1, the default value is 01. If SEGMENT=2, the default value is (01,02).

Note: *seg2_loc_mod_addr* is valid only if SEGMENT=2.

VTAM uses this operand only when processing DR ADD definition statements for PUs that support LPDA2.

LNCTL



statements: GROUP
dependency: VTAM and NCP

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.

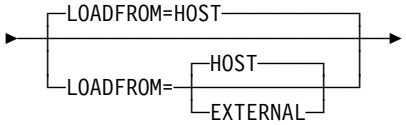
LNCTL=CA
indicates that the definition statements following this GROUP definition statement identify intermediate routing node and boundary node channel definitions.

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



statements: PCCU
dependency: see notes

specifies whether VTAM loads the NCP module from the host or from the communication controller external disk storage.

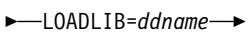
LOADFROM=HOST
loads the NCP from the host.

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

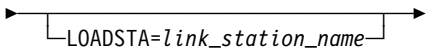


statements: BUILD
dependency: none

specifies the data set or file in which the NCP resides.

MVS Name the data set on a DD statement in the VTAM start procedure. **VM** Issue a FILEDEF command for *ddname* before you load the NCP.

LOADSTA



statements: PCCU
dependency: 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 “Channel-Attached NCP” in the *VTAM 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 channel unit address and adding “-S.” If the channel unit address is

3-digits, a leading zero is added to the name. For example, the channel unit address 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 “VARY ACT Command” in *VTAM Operation*.

LOCADD

```
►—LOCADD=address—►
```

statements: LINE
dependency: 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. LOCADDR is required when you specify physical resources on the GROUP definition statement.

Use the following format when coding LOCADD:

```
LOCADD=aabbcccccccccccc
```

where:

aa is the token-ring interface coupler (TIC) number of the communication controller (in the range 00–99). The TIC number equates with the PORTADD keyword on the NCP LINE definition statement.

bb is the service access point (SAP) address of the terminal. It must be a multiple of 4.

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
dependency: VTAM and NCP
range: 0–255 PU Types 2 and 2.1
0–63 PU Type 1

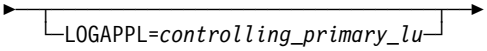
specifies the logical unit’s local address at the associated physical unit. Specify the addresses in ascending order. LOCADDR=0 specifies an independent LU.

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



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; NetView or TSO (MVS)

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.

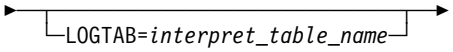
Automatic logon LU-LU sessions are not redriven after CP-CP outages even though the CP-CP sessions might redrive.

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 (MVS) program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery situations.

For more information on establishing sessions, see “Automatic Logons” in the *VTAM Network Implementation Guide*.

LOGTAB

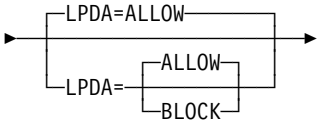


statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; dependent LUs only

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 690.

LOGTAB is valid for dependent logical units only.

LPDA



statements: PU
dependency: VTAM and NCP

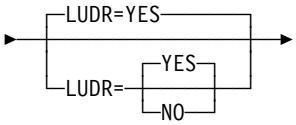
specifies whether Link Problem Determination Aid (LPDA) tests can be run for the physical unit.

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



statements: GROUP, LINE, PU, LU
dependency: 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.

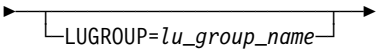
LUDR=NO
indicates that VTAM rejects any attempt to move or delete the LU.

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.

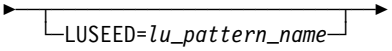
LUGROUP



statements: PU
dependency: 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



statements: PU
dependency: VTAM only; LUGROUP

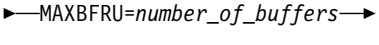
provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU. LUSEED is valid only when you code the LUGROUP operand.

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 “Selection of Definitions for Dependent LUs (SDDLU) Exit Routine” in *VTAM Customization*.

MAXBFRU



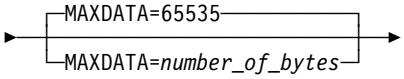
statements: HOST
dependency: none
range: 1–255

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 "Sharing NCP Peripheral Nodes (CMC Configurations)" in the *VTAM Network Implementation Guide* for more information on CMC configurations.

MAXDATA



statements: PCCU
dependency: none

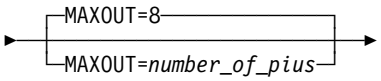
specifies the maximum number of bytes that the NCP can receive in one PIU or PIU segment, including the transmission header (TH) and the request/response header (RH).

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 "HOST Definition Statement" in the *VTAM 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 95 for a description of the MAXDATA operand in a channel-attachment major node.

MAXOUT

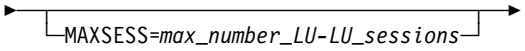


statements: GROUP, LINE, PU
dependency: none

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



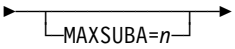
statements: GROUP, LINE, PU, LU
dependency: VTAM and NCP; independent LU
range: 1–65535

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



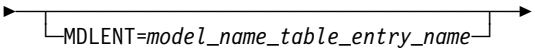
statements: BUILD, NETWORK
dependency: ENA
range: 1–255

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 564. Refer to “How VTAM Handles Network and Subarea Addressing” in the *VTAM Network Implementation Guide* for considerations when coding this operand.

MDLENT

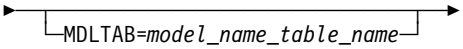


statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; MDLTAB

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



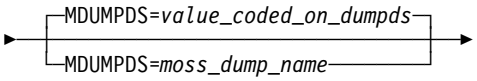
statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only

specifies the name of the model name table to be used for a logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

MDUMPDS



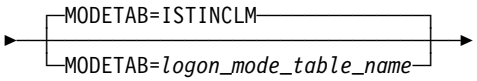
statements: PCCU
dependency: none

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 312.

MVS *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. **VM** *dump_name* must be on a FILEDEF statement. **VSE** Use the device name (SYSxxx) that is coded on the EXTENT or ASSIGN statement for the DLBL statement that contains the file name for the dump dataset.

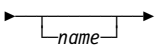
MODETAB



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: 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 666. 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



statements: PCCU, BUILD, HOST, NETWORK, FRSESET

statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL

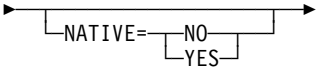
assigns a name to the resource being defined by the definition statement.

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



statements: GROUP, LINE, PU
dependency: APPN only; BN=YES

specifies whether this link station represents a connection to a native node.

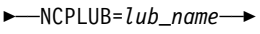
If NATIVE is not specified on the ADJCP definition statement or on the PU definition statement, the two nodes negotiate their subnetwork affiliation during connection establishment.

- 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.

NCPLUB (VSE)

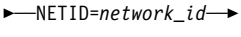


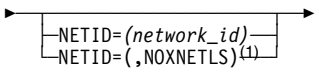
statements: PCCU
dependency: none

is the name of the NCP load file that contains the NCP phase. The device name appears on the DLBL statement for the NCP phase.

For more information on the NCP load file, refer to the *VTAM Network Implementation Guide*.

NETID





Note:
[1] NOXNETLS applies only to the PU definition statement.

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 page 600 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.

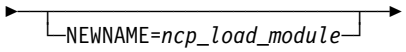
NETID=(,NOXNETLS)

PU only: (,NOXNETLS) specifies that VTAM does not allow nonnative network attachment for LU-LU sessions for this PU. When NETID=(,NOXNETLS) is specified, VTAM indicates to the NCP that this PU will use VTAM's native network ID.

LU-LU sessions should fail if the NETID sent in the BIND from the LU is different than the NETID specified in VTAM, even though PU-PU connections can establish. If no NETID is included in the BIND, or if the NETID 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.

NEWNAME



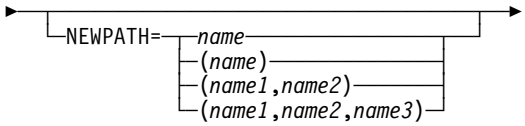
statements: BUILD
dependency: none

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 “VARY ACT Command” in *VTAM Operation* for information on LOADMOD.

NEWPATH



statements: PCCU
dependency: 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.

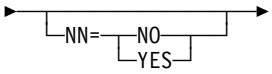
Up to three member names can be specified; if more than three member names are specified on the PCCU definition statement, only the first three are used.

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 “VARY ACT Command” in *VTAM 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



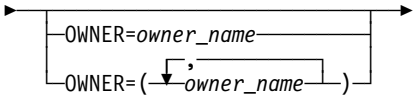
statements: GROUP, LINE, PU
dependency: APPN only

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



statements: PCCU, GROUP, LINE, PU
dependency: 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.

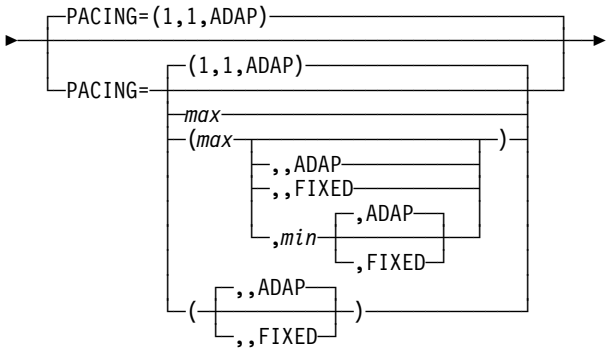
GROUP, LINE, PU: associates the resources in a communication controller with the VTAM that controls these resources. OWNER associates those resources with the PCCU definition statement that has the same OWNER value. A resource can be associated with several VTAMs. If you do not specify the OWNER operand, 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



statements: GROUP, LINE, PU, LU, CLUSTER
dependency: 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).

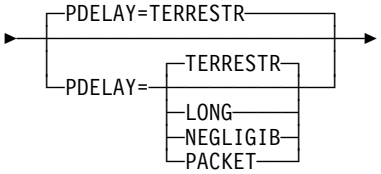
type is used only by NCP, and it requires at least NCP V5R4.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

PDELAY



statements: GROUP, LINE, PU
dependency: 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.

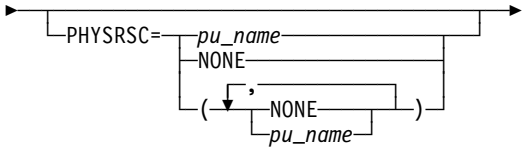
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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



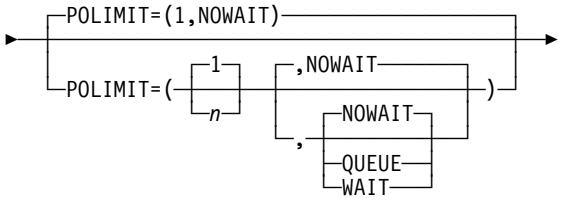
statements: GROUP
dependency: none

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



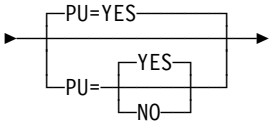
statements: GROUP, LINE
dependency: VTAM and NCP

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



statements: GROUP, LINE
dependency: VTAM only

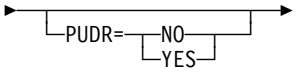
specifies whether a BSC 3270 is treated as a physical unit and whether all terminals subordinate to the cluster are treated as logical units.

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

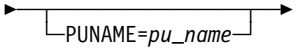


statements: GROUP, LINE, PU
dependency: VTAM and NCP

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

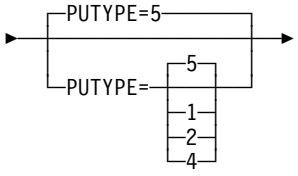


statements: BUILD
dependency: none

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



statements: GROUP, LINE, PU
dependency: VTAM and NCP

specifies the physical unit type represented by the PU definition statement.

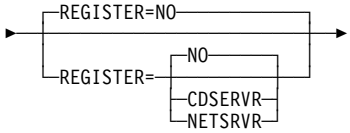
Notes:

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 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*.

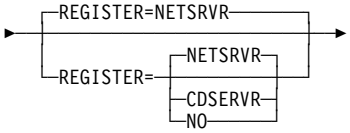
REGISTER

For Independent LUs:



statements: PU, LU
dependency: APPN only

For Dependent LUs:



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.

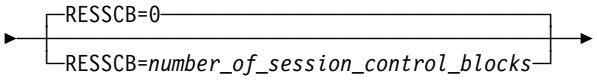
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

RESSCB

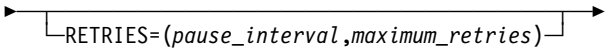


statements: LU
dependency: VTAM and NCP
range: 0–65535

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



statements: GROUP, LINE, PU
dependency: VTAM and NCP
range: 0–255

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*.

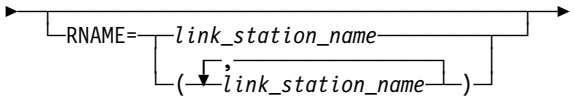
maximum_retries
specifies the maximum number of retry sequences. Valid values are 0–127.

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 *n* 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



statements: PCCU
dependency: none

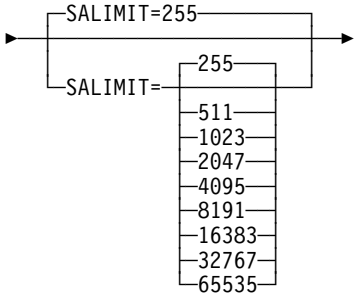
must match the label on the PU definition statement that describes this NCP in an adjacent communication controller or host. (That PU definition statement identifies, to the other NCP or host, the link that connects it to this communication controller.) You can specify up to 13 SDLC or channel link station 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 “Channel-Attached NCPs in a CMC Configuration” in the *VTAM Network Implementation Guide*.

SALIMIT

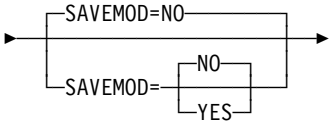


statements: BUILD, NETWORK
dependency: none

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



statements: PCCU
dependency: 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.

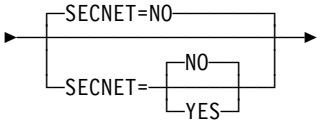
SAVEMOD=YES
If you code YES, the NCP load module is saved after it is loaded from the host.

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 “VARY ACT Command” in *VTAM Operation*.

SECNET



statements: LINE, PU
dependency: 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.

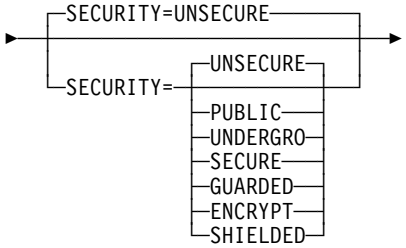
This operand applies only if you have IBM 3710 Network Controllers or IBM 3174 token-ring attachments in your network.

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 (MVS, VM)



statements: GROUP, LINE, PU
dependency: 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.

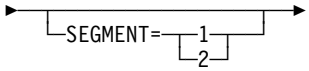
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SEGMENT



statements: PU
dependency: 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.

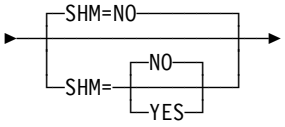
SEGMENT=2
indicates the station is on segment number 2. This is the default value if TAILING=YES is specified.

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



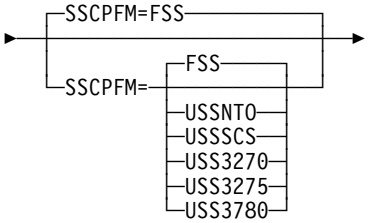
statements: GROUP
dependency: VTAM only

specifies whether short-hold mode is supported. Only lines representing switched virtual circuits (SVCs) controled 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



statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: 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
is 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.

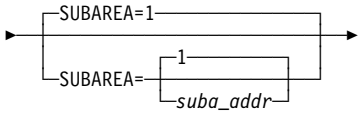
SSCPFM=USS3270
allows printing of a user-defined logon message (USSMSG10) on BSC printers. Code SSCPFM=USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE=1) or SDLC 3275 (PUTYPE=1).

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.

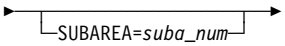
Note: Coding SSCPFM=USS3275 prevents the local copy hardware function key from working with certain BSC printers. To make sure the local copy function works, code SSCPFM=USS3270.

SSCPFM=USS3780
is 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.

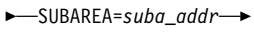
SUBAREA



statements: HOST
dependency: none



statements: PCCU
dependency: none



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 “NCP Definition Statements for VTAM” in the *VTAM 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.

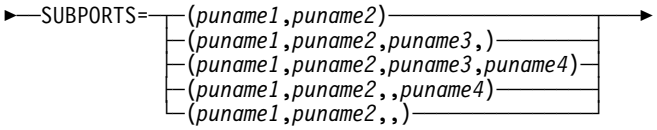
This operand is required.

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.

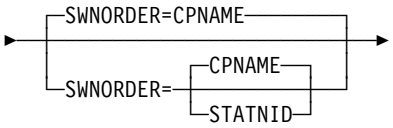
SUBPORTS



statements: FRSESET
dependency: 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



statements: GROUP, LINE
dependency: none

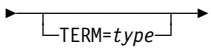
specifies the way VTAM locates a switched PU. If you do not code SWNORDER, VTAM searches for a PU by the CPNAME first. If VTAM does not find the PU by the CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

This operand overrides the SWNORDER start option value.

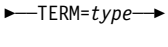
SWNORDER=CPNAME
specifies that VTAM searches for a switched PU by the CPNAME first.

SWNORDER=STATNID
specifies that VTAM searches for a switched PU by the station identifier first.

TERM



statements: PU, LU
dependency: VTAM and NCP



statements: GROUP, LINE, CLUSTER, TERMINAL
dependency: VTAM and NCP

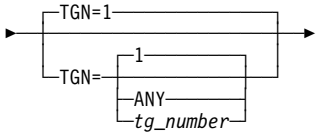
describes, for a virtual logical unit supported through Network Terminal Option (NTO), the device data stream compatible characteristics. These characteristics are placed in an 8-byte program-supplied storage area, after an INQUIRE DEVCHAR macroinstruction is executed by an application program that specifies the virtual logical unit name. For more information on the DEVCHAR DSECT, refer to "Device Characteristics Field" in *VTAM 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.

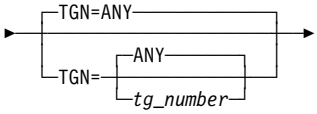
TGN

For subarea TGN:



statements: PCCU, PU
dependency: VTAM and NCP
range: 1–255

For APPN TGN:



statements: PCCU, PU
dependency: VTAM and NCP; CPNAME; NETID
range: 0–21

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.

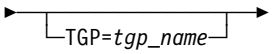
For APPN, coding TGN=0 means that the PU does not support parallel TGs.

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

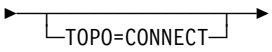


statements: GROUP, LINE, PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO



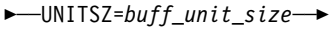
statements: PU
dependency: APPN only

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.

UNITSZ

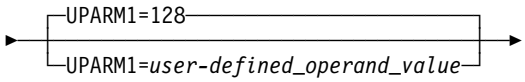


statements: HOST
dependency: *bufsize* on IOBUF start option
range: 64–4000 bytes

specifies the size of the buffer units used for data transfers from the NCP. For information on specifying the value of UNITSZ, refer to “Guidelines for Setting UNITSZ” in the *VTAM Network Implementation Guide*.

Note: VTAM uses the value specified by *bufsize* on the IOBUF start option to determine buffer size for data transfer from the NCP.

UPARM1

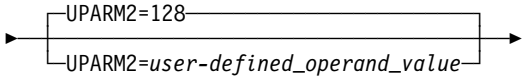


statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

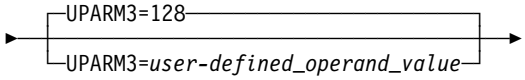
UPARM2



statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

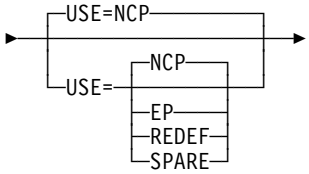
UPARM3



statements: GROUP, LINE, PU
dependency: APPN only

specifies a user-defined operand value.

USE



statements: GROUP, LINE
Note: code USE=REDEF, USE=SPARE only on LINE.
dependency: VTAM and NCP

specifies the initial operating condition of the line for lines that can operate alternately in network control mode and emulation mode.

VM, VSE 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 VM,VSE
specifies that the line is being used, but can be changed to a spare line. The mode of operation is assumed to be NCP.

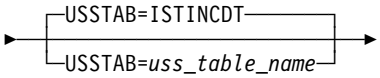
USE=REDEF is used with the MODIFY command to change it from a defined line to a spare line and vice versa.

Note that a line defined with USE=REDEF must be the only line in a line group.

USE=SPARE VM,VSE
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



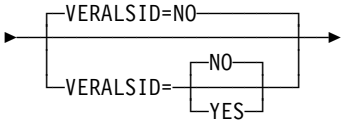
statements: GROUP, LINE, PU, LU, CLUSTER, TERMINAL
dependency: VTAM only; dependent LUs only

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 690.

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



statements: GROUP, LINE, PU
dependency: APPN only

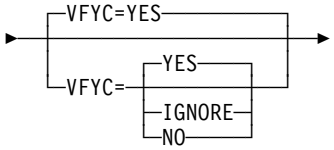
specifies whether the adjacent link station name should be used to determine the link station.

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



statements: PCCU
dependency: 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.

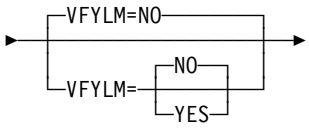
VFYC=NO
specifies that VTAM reloads the communication controller.

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



statements: PCCU
dependency: none

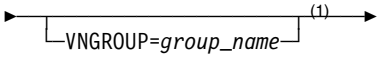
specifies how VTAM handles mismatches between the name and subarea of the load module currently in the NCP and the NEWNAME operand on the BUILD definition statement. When the names or subareas do not match, one of the following actions occurs:

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. (AUTOSYN is described on page 303.) on the VARY ACT command for an NCP.

VNGROUP



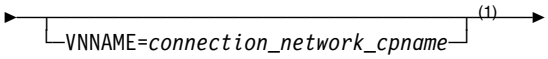
Note:
[1] Code VNGROUP if you code VNNAME.

statements: GROUP, LINE
dependency: 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



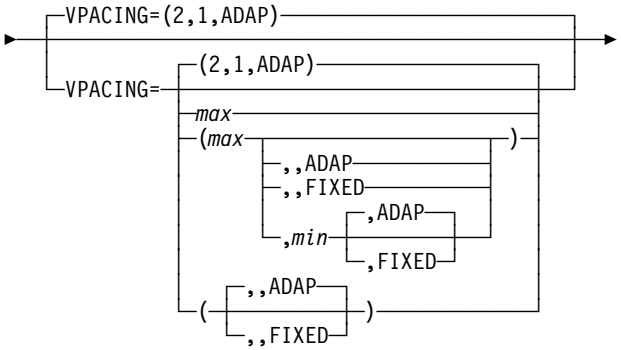
Note:
[1] Code VNNAME if you code VNGROUP.

statements: GROUP, LINE
dependency: 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



statements: GROUP, LINE, PU, LU, CLUSTER
dependency: 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).

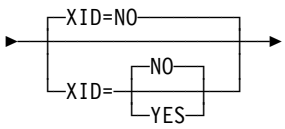
type is used only by NCP, and it requires at least NCP V5R4.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

XID



statements: GROUP, LINE, PU
dependency: VTAM and NCP

specifies whether a physical unit on a nonswitched SDLC line can identify itself to VTAM through XID.

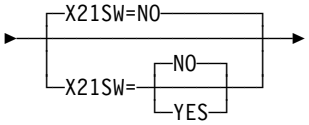
XID=NO
indicates an XID is not sent to the physical unit.

For 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.

X21SW (VSE)



statements: GROUP
dependency: VTAM and NCP

specifies whether the lines in the group are X.21 switched lines.

X21SW=NO
specifies that the lines in a group are not X.21 switched lines.

X21SW=YES
specifies that the lines in a line group are X.21 switched lines. VTAM adds the required end-of-number (EON) character to the dial digits for an auto-call operation or permits a direct call switched path to be used.

Refer to the *NCP, SSP, and EP Resource Definition Reference* for proper coding of the X21SW operand.

Packet Major Node (VM, VSE)

To define a packet major node for each channel unit address pair that is configured as an X.25 port on your integrated communication adapter, code definition statements to specify the following characteristics:

- The node type (VBUILD definition statement)
- The X.25 port (PORT definition statement)
- Flow control parameters for virtual circuits (VCPARMS definition statement)
- The line group (GROUP definition statement)
- The lines (LINE definition statement)
- The physical units (PU definition statement) that can be connected over the lines (SNA only)
- The logical units (LU definition) under the physical units (SNA only).

Refer to the *VTAM Network Implementation Guide* for information on and coding examples for packet major nodes.

Nonswitched Packet Major Node

A permanent virtual circuit appears to VTAM as an SDLC nonswitched line. For each permanent virtual circuit to which you have subscribed, code an SDLC nonswitched line.

Note: Whenever "nonswitched line" appears in this section, it refers to the line representing the permanent virtual circuit to VTAM.

This section contains all the definition statements with operands needed to code a nonswitched line in the packet major node. In this section, you will find the GROUP, LINE, PU, and LU definition statements.

VBUILD Definition Statement: See "VBUILD Definition Statement" on page 363 for details on the VBUILD definition statement.

PORT Definition Statement: See "PORT Definition Statement" on page 363 for details on the PORT definition statement.

VCPARMS Definition Statement: See "VCPARMS" on page 363 for details on the VCPARMS definition statement.

GROUP Definition Statement: This GROUP definition statement defines a nonswitched line group. It also carries other operands that sift down to following definition statements that do not explicitly override them.

Note: VM You must define a nonswitched line group for operation of VTAM Common Network Services (VCNS).

LINE Definition Statement: Code one LINE definition statement for each nonswitched line. The LINE definition statement defines to VTAM the following characteristics:

- The name for the line
- The logical channel identifier for the permanent virtual circuit represented by this line
- Certain procedural options to be used for this line.

The line identified by the ADDRESS operand on this definition statement may be defined only once under a packet major node. This means that you can use the default for the ADDRESS operand only once in a packet major node.

PU Definition Statement: Code one PU definition statement for the physical unit of any type (1, 2, 4, or 5) with which VTAM communicates over this nonswitched line. The PU definition statement specifies the following characteristics:

- The resource name for the physical unit
- The physical unit type (1, 2, 4, or 5)
- The station address of the physical unit
- The maximum size of a PIU or PIU segment that VTAM will send to the physical unit
- The subarea address of the physical unit (types 4 and 5 only).

Notes:

1. VTAM uses the PU definition statement to define the link station for an adjacent host processor (physical unit type 5) or communication controller (physical unit type 4).
2. Users of VTAM Common Network Services (VCNS) must not code PU definition statements following a LINE definition statement that specifies USER=VCNS.

LU Definition Statement: Code one LU definition statement for each logical unit associated with a physical unit (type 1 or 2) attached to a nonswitched line. LU definition statements immediately follow the PU definition statement for the physical unit with which they are associated.

The LU definition statement defines the following characteristics:

- The resource name for the logical unit
- The local address for the logical unit
- Use of the pacing option.

Note: Users of VCNS must not code LU definition statements following a LINE definition statement that specifies USER=VCNS.

Table 34 (Page 1 of 3). Packet Major Node Definition Statements and Operands - Nonswitched

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=PACKET	396	R			
		CONFGDS	371				VSE
		CONFGPW	371				VSE CONFGDS
name	PORT	CUADDR	373	R			
		NETTYPE	385	R			
		CHARGACC	370			NO	SNA connections; reverse charging acceptance; NETTYPE
		CHARGE	371			NO	SNA connections; NETTYPE

Table 34 (Page 2 of 3). Packet Major Node Definition Statements and Operands - Nonswitched

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		DIALNO	374				
		MAXOUT	381			7	
		NETLEVEL	384			80	
		PLENGTH	389			128	see description
		PMOD	390			8	NETTYPE
		PWINDOW	390			2	see description
		REPLYTO	392			3.0	
		RETRIES	392			7	
		VCALLS	398				required for switched virtual circuits; NETTYPE
name	VCPARMS	LC	377				
		PLENGTH	389			128	
		PWINDOW	390			2	
name	GROUP	DIAL=NO	373				nonswitched lines
		LNCTL=SDLC	379				
		SPAN	393				NetView
name	LINE	ADDRESS	368			001 VM 030 VSE	
		ISTATUS	377		G	ACTIVE	
		SPAN	393				NetView
		USER	397			SNA	VM ;see description
name	PU	ADDR	367	R			PU types 1, 2
		AUTHLEN	369			YES	APPN only PU Type 2.1
		CAPACITY	370		G,L	8K	APPN only
		CONNTYPE	372		G,L	CONNTYPE start option	APPN only; NODETYPE
		COSTBYTE	372		G,L	0	APPN only
		COSTTIME	372		G,L	0	APPN only
		CPCP	373		G,L	CPCP start option	APPN only
		CPNAME	373				
		DISCNT	374		G,L	(NO)	PU types 1, 2
		DYNADJCP	375		G,L	DYNADJCP start option	APPN only; CPNAME
		DYNLU	375		G,L	DYNLU start option	PU types 1, 2
		ISTATUS	377		G	ACTIVE	
		LIMRES	378		G,L	NO	LU 6.2 only
		LUGROUP	380				
		LUSEED	380				LUGROUP
		MAXDATA	381		G,L	261 (PU type 1) 265 (PU type 2)	PU types 1, 2
		NATIVE	383				BN=YES; APPN only
		NETID	384				PU types 1, 2
		NN	387		G,L		APPN only

Table 34 (Page 3 of 3). Packet Major Node Definition Statements and Operands - Nonswitched

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		PDELAY	388		G,L	TERRESTR	APPN only
		PUTYPE	390		G,L	2	
		SECURITY	392		G,L	UNSECURE	APPN only VM
		SPAN	393				NetView
		SUBAREA	394				PUTYPE=4 or 5
		TGN (subarea)	395			1	PU types 4, 5
		TGN (APPN)	395			ANY	APPN only; PU type 2
		TGP	396		G,L		APPN only
		TOPO	396			CONNECT	APPN only
		UPARM1	396		G,L	128	APPN only
		UPARM2	397		G,L	128	APPN only
		UPARM3	397		G,L	128	APPN only
		VERALSID	399		G,L	NO	APPN only
		XID	400		G,L	NO	type 2.1
name	LU	LOCADDR	379	R			see description
		ASLENT	368		G,L,P		PU types 1, 2; ASLTAB
		ASLTAB	369		G,L,P		PU types 1, 2
		DLOGMOD	375		G,L,P		PU types 1, 2
		EAS	377		G,L,P	256	independent LUs
		ISTATUS	377		G	ACTIVE	
		LOGAPPL	379		G,L,P		PU types 1, 2; NetView
		LOGTAB	380		G,L,P		dependent LUs
		MDLENT	382		G,L,P		PU types 1, 2; MDLTAB
		MDLTAB	382		G,L,P		PU types 1, 2
		MODETAB	382		G,L,P	ISTINCLM	PU types 1, 2
		PACING	387		G,L,P	(1,1,ADAP)	PU types 1, 2
		REGISTER	391		P	see description	APPN only
		SPAN	393				NetView
		SSCPFM	393		G,L,P	FSS	PU types 1, 2
		USSTAB	398		G,L,P	ISTINCDT	dependent LUs
		VPACING	399		G,L,P	(2,1,ADAP)	PU types 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.

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.

Switched Packet Major Node

A switched virtual circuit appears to VTAM as an SDLC switched line. However, there is no one-to-one relationship with the switched virtual circuit you coded on the VCALLS operand. Rather, the mapping of a switched virtual circuit to a particular switched line occurs when a connection is made.

Note: Whenever "switched line" appears in this section, it refers to the line representing the switched virtual circuit to VTAM.

This section contains all the definition statements with operands needed to code a switched line of the packet major node. Those include the GROUP, LINE, and PU definition statements.

VBUILD Definition Statement: See "VBUILD Definition Statement" on page 363 for details on the VBUILD definition statement.

PORT Definition Statement: See "PORT Definition Statement" on page 363 for details on the PORT definition statement.

VCPARMS Definition Statement: See "VCPARMS" on page 363 for details on the VCPARMS definition statement.

GROUP Definition Statement: This GROUP definition statement defines a switched line group; it also carries other operands that sift down to following definition statements that do not explicitly override them.

If you define a line in more than one line group, only one of the groups can be active.

LINE Definition Statement: Code one LINE definition statement for each switched line. The LINE definition statement defines the following characteristics to VTAM:

- The name for this line
- Certain procedural options to be used for this line.

PU Definition Statement: Code one PU definition statement for each physical unit with which VTAM communicates over this switched line.

The PU definition statement defines the following:

- The resource name for the physical unit
- The maximum number of logical units to be associated with the physical unit.

Table 35 (Page 1 of 2). Packet Major Node Definition Statements and Operands - Switched

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=PACKET	396	R			
		CONFGDS	371				VSE
		CONFGPW	371				VSE CONFGDS
name	PORT	CUADDR	373	R			
		NETTYPE	385	R			

Table 35 (Page 2 of 2). Packet Major Node Definition Statements and Operands - Switched

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		CHARGACC	370			NO	SNA connections; reverse charging acceptance
		CHARGE	371			NO	SNA connections
		DIALNO	374				
		MAXOUT	381			7	
		NETLEVEL	384			80	
		PLENGTH	389			128	
		PMOD	390			8	
		PWINDOW	390			2	
		REPLYTO	392			3.0	
		RETRIES	392			7	
		VCALLS	398				required for switched virtual circuits
name	VCPARMS	LC	377				
		PLENGTH	389			128	
		PWINDOW	390			2	
name	GROUP	DIAL=YES	373				switched lines
		DYNPU	376			NO	
		DYNPUPFX	376			CN	DYNPU=YES
		LNCTL=SDLC	379				
		SPAN	393				NetView
		SUBADIAL	394			NO	
name	LINE	ANSWER	368		G	ON	
		CALL	369		G	IN	VCALLS
		ISTATUS	377		G	ACTIVE	
		LIMRES	378		G	NO	LU 6.2 only
		SPAN	393				NetView
		SWNORDER	394		G	CPNAME	
name	PU	AUTHLEN	369			YES	APPN only PU Type 2.1
		ISTATUS	377		G	ACTIVE	
		SPAN	393				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.

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 channel unit address pair configured as an X.25 port on your integrated communication adapter.

Any number of VBUILD TYPE=PACKET definition statements up to the number of X.25 ports supported by the host processor can be coded and filed. Each channel unit address pair of the subsystem or adapter, which is configured as an X.25 port, requires a separate VBUILD TYPE=PACKET definition deck to be filed.

Note for users of the NetView program: To provide a unique address for the NetView program to display alert and statistical data associated with the physical attachment to the network, the VBUILD TYPE=PACKET major node definition must have certain specifications. The first GROUP and LINE definition statements under the VBUILD major node definition must be for a nonswitched line with a physical unit representing a permanent virtual circuit (PVC). Refer to the *VTAM Network Implementation Guide* for information on providing a unique address for displaying alert and statistical data.

PORT Definition Statement: The PORT definition statement identifies the X.25 port. It specifies the X.25 link level options, network type, network level, switched virtual circuit definitions, and options selected for all virtual circuits on that port.

Code one PORT definition statement for each VBUILD TYPE=PACKET definition statement. The PORT definition statement must immediately follow the VBUILD definition statement.

The PORT definition statement causes the LNCTL operand of the GROUP definition statements that follow to default to LNCTL=SDLC.

VCPARMS: VCPARMS is an optional definition statement that, if specified, immediately follows the PORT definition statement under a packet major node. You can code up to seven VCPARMS definition statements for each packet major node.

For permanent virtual circuits: The VCPARMS definition statement specifies values for packet and window sizes other than the standard values for each permanent virtual circuit or group of permanent virtual circuits for which the VCPARMS definition statement is coded.

For ES/9370*, 9221, and 9371 processors, when NETTYPE=DTE or NETTYPE=DCE, the values you code for PLENGTH and PWINDOW must match the PLENGTH and PWINDOW values defined for the same logical channels of the remote DTE.

For switched virtual circuits: The VCPARMS definition statement specifies values used to negotiate requested flow control parameters on incoming calls. A user who requested the optional user facility, Flow Control Parameter Negotiation, can specify VCPARMS to negotiate the packet and window sizes when the switched virtual circuits are set up. When a remote DTE calls, the network indicates the packet and window size it proposes for each direction of the transmission. The values in the VCPARMS definition statement are used to tell the network the values your installation accepts. The values for packet size must be between the value subscribed to and the standard value of 128. The value of

window size must be between the standard value of 2 and the window size subscribed to. The VCPARMS definition statement applies to SNA switched virtual circuits only.

To negotiate flow control parameters, you must have subscribed to the optional user facility, Flow Control Parameter Negotiation. Additionally, your network must accept Call Accepted packets with facility fields. Negotiating packet and window values enables your installation to limit the amount of storage used by connections established by incoming calls. Negotiating flow control parameters begins differently for different networks. For details about your network, consult your network documentation.

The values defined by the PLENGTH and PWINDOW operands on the PORT definition statement are always used when the call originates from your host.

Dynamic Change of Operands: You can dynamically change the following 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
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.

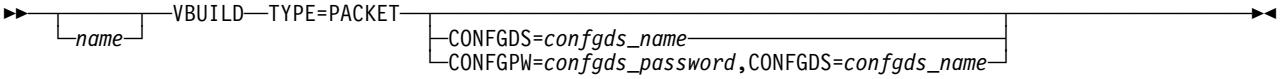
You can dynamically change the operands associated with tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) at any time without inactivating resources.

Refer to “Dynamic Reconfiguration and Change of Operands” in the *VTAM Network Implementation Guide* and “VARY ACT Command” in *VTAM Operation* for more information on the VARY ACT,UPDATE=ALL command.

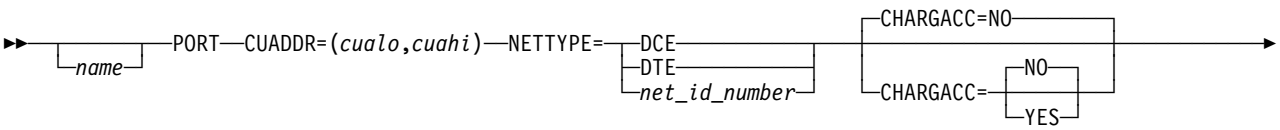
Full Syntax

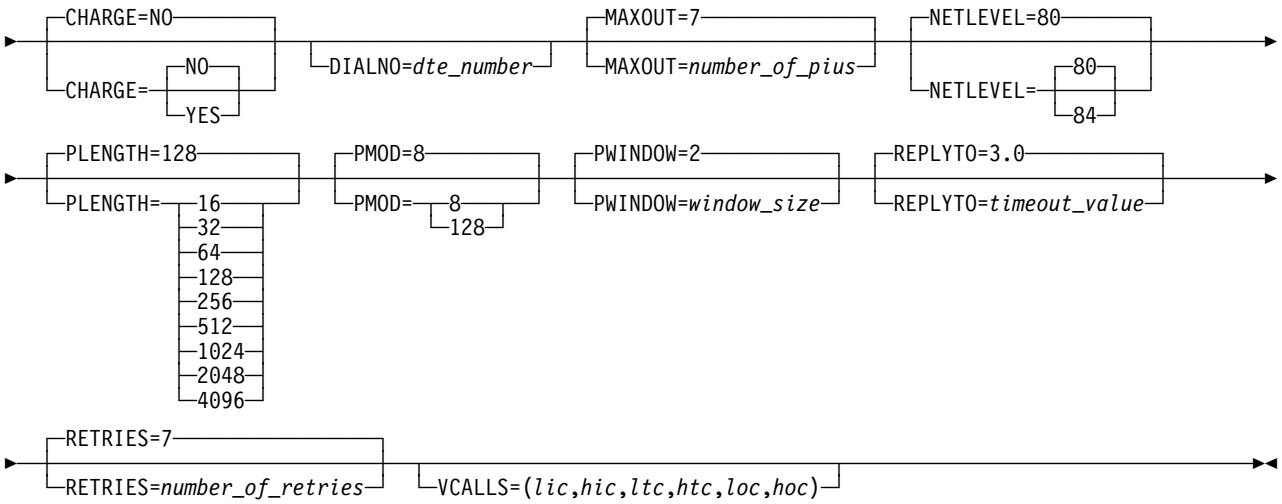
The full syntax for the packet major node follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD

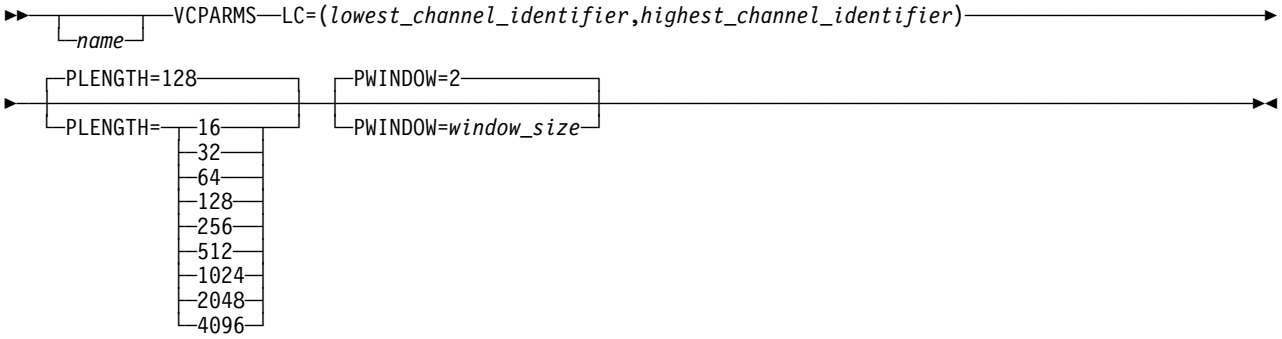


PORT

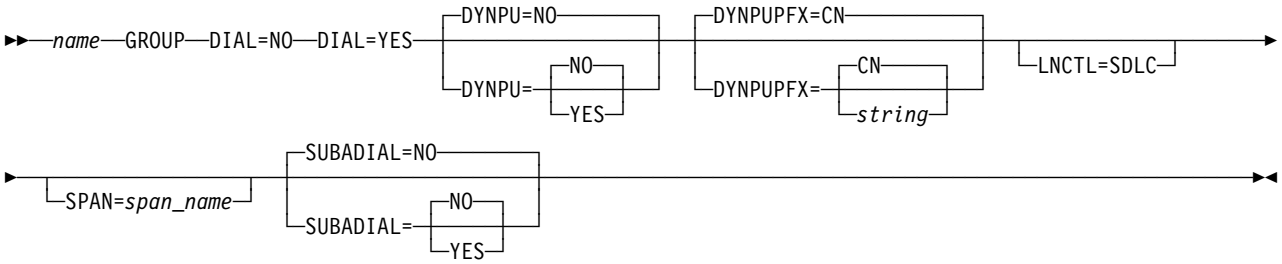




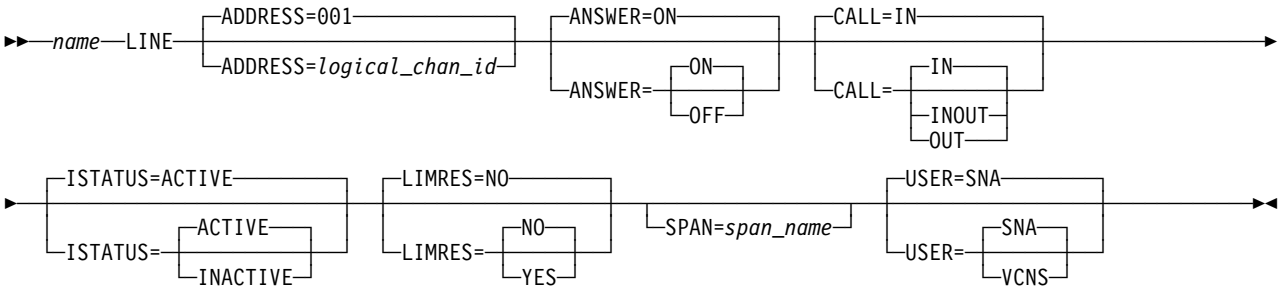
VCPARMS



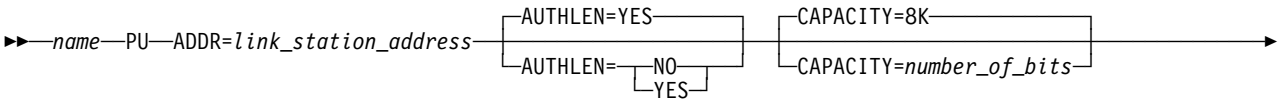
GROUP

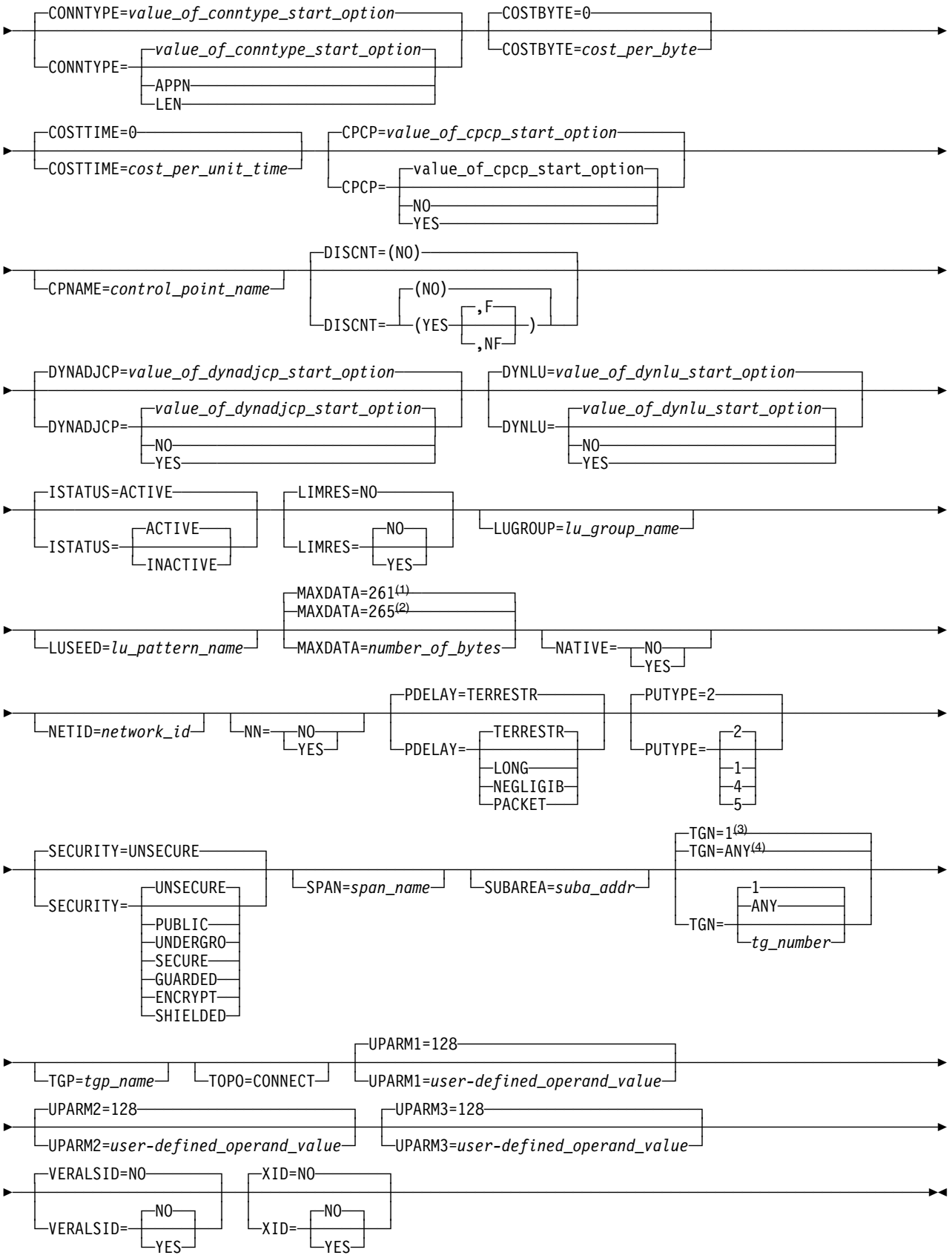


LINE



PU





Notes:

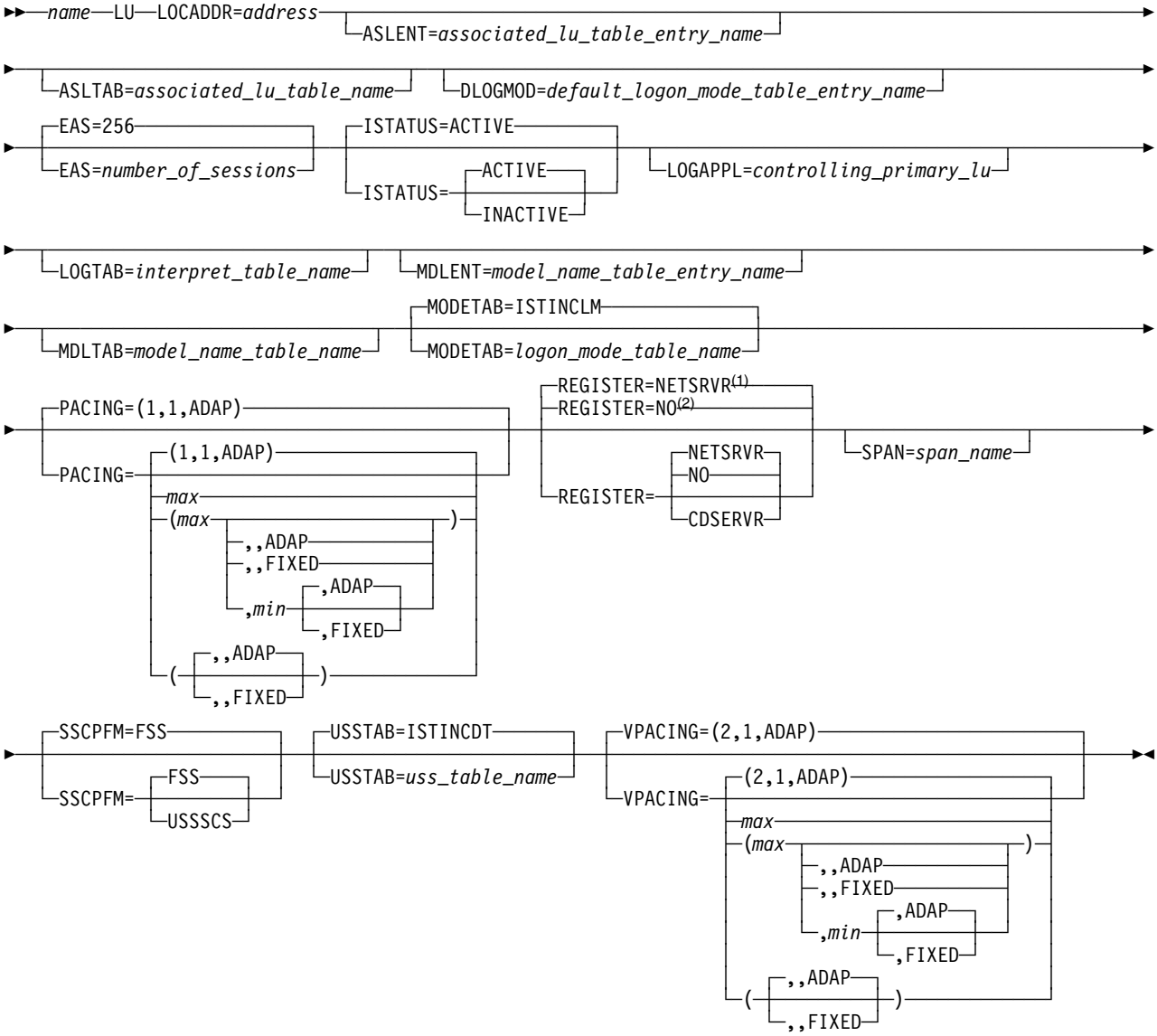
1 MAXDATA defaults to 261 when PUTYPE=1 is coded.

2 MAXDATA defaults to 265 when PUTYPE=2 is coded.

[3] TGN defaults to 1 for subarea.

[4] TGN defaults to ANY for APPN.

LU



Notes:

[1] REGISTER defaults to NETSRVR for dependent LUs.

[2] REGISTER defaults to NO for independent LUs.

Operand Descriptions

ADDR

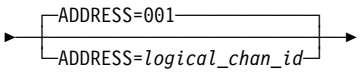
statements: PU
dependency: applies only to PU types 1 and 2
range: X'01'–X'FE'.

specifies the 2-digit hexadecimal station address which identifies the physical unit at the logical link control level. If the station address is X'B0', code ADDR=B0. Consult the relevant device publication for guidance in assigning this value.

Note: This operand is required for type 1 and 2 physical units. If you code PUTYPE=4 or PUTYPE=5, and ADDR, ADDR defaults to C1.

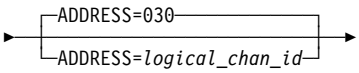
ADDRESS

For VM



statements: LINE
dependency: none
range: X'000'–X'FFF'

For VSE

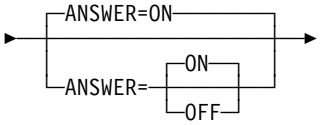


statements: LINE
dependency: none
range: X'000'–X'FFF'

specifies a 3-digit hexadecimal logical channel identifier for the permanent virtual circuit.

The operator cannot override the defined address with the U operand of the VARY ACT command. See *VTAM Operation* for details on using the U operand.

ANSWER



statements: GROUP, LINE
dependency: none

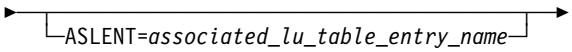
specifies whether lines can be used for incoming calls. This operand is ignored for switched subarea connections.

ANSWER=OFF
specifies that the physical units cannot dial in to VTAM, regardless of whether the line is active or inactive.

ANSWER=ON
specifies that the 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. See *VTAM Operation* for information on using the VARY ANS command.

ASLENT

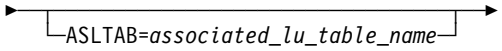


statements: GROUP, LINE, PU, LU
dependency: ASLTAB; applies only to PU types 1 and 2

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 table named in ASLTAB by default.

ASLTAB



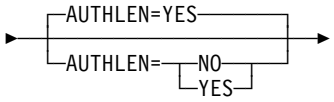
statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2

specifies the name of the associated LU table to be used for the logical unit. See "Associated LU Table" on page 638 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 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 714 for more information on PRINTER1 and PRINTER2 operands of the LOGON command.

The operator can use the MODIFY TABLE command to dynamically replace an associated LU table (ASLTAB). See *VTAM Operation* for more information about the MODIFY TABLE command.

AUTHLEN



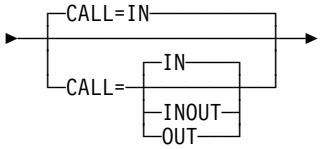
statements: PU
dependency: PU Type 2.1; APPN only

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.

CALL



statements: GROUP, LINE
dependency: VCALLS

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 call VTAM).

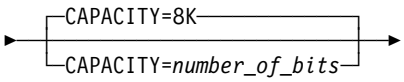
CALL=INOUT
specifies that the line is to be used for both incoming and outgoing calls.

CALL=OUT
specifies that the line is to be used only for outgoing calls (VTAM calls the physical units).

The VCALLS operand specification is related to the specification of this operand as follows:

- If you specified *lic* and *hic* or *ltc* and *htc* for VCALLS, you must specify CALL=IN or CALL=INOUT on at least one LINE definition statement, for VTAM to accept incoming calls.
- If you specify CALL=OUT or CALL=INOUT, you must specify *loc* and *hoc* or *ltc* and *htc*, for VCALLS for VTAM to dial out.

CAPACITY

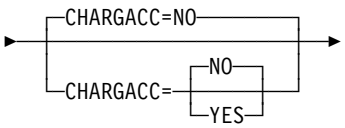


statements: GROUP, LINE, PU
dependency: APPN only
range: 1K–1000M

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

CHARGACC



statements: PORT
dependency: applies only to SNA connections;
requires reverse charging acceptance facility; NETTYPE

specifies whether incoming calls with reverse charging will be accepted or rejected by VTAM. CHARGACC applies to all switched SNA virtual circuits on this port with incoming call capability.

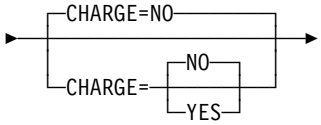
CHARGACC=YES
specifies that reverse charging requests will be honored.

CHARGACC=NO
specifies that reverse charging acceptance will not be honored.

Specify CHARGACC only if you subscribe to the reverse charging acceptance optional user facility.

For ES/9370, 9221, and 9371 processors, when NETTYPE=DTE or NETTYPE=DCE, this operand is ignored.

CHARGE



statements: PORT
dependency: applies only to SNA connections; NETTYPE

specifies the charging method for outgoing call requests established through this port.

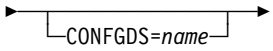
CHARGE=NO
specifies that outgoing calls on this port do not request reverse charging. This port will be charged for the call.

CHARGE=NO can be overridden by the DIALNO operand on the PATH definition statement, which is explained on page 421.

CHARGE=YES
specifies that all outgoing calls through this port request reverse charging by indicating the optional user facility, reverse charging, in the call request packet.

For ES/9370, 9221, and 9371 processors, when NETTYPE=DCE or NETTYPE=DTE, this operand is ignored.

CONFGDS (VSE)

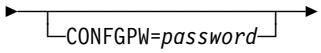


statements: VBUILD
dependency: none

is a 1–8 character file name that identifies the configuration restart file defined by the user for this major node.

Include in the VTAM start procedure a DLBL statement (MVS) that has this data definition name.

CONFGPW (VSE)

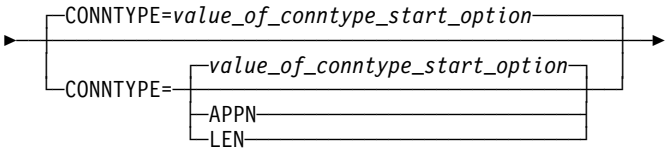


statements: VBUILD
dependency: CONFGDS

specifies the 1–7 character alphanumeric password, if required, for VTAM to gain access to the configuration restart file. If CONFGPW is not specified, but is required by VSAM, VSAM prompts for the correct password when VTAM attempts to open the file.

You can specify this operand only if the CONFGDS operand is specified.

CONNTYPE



statements: GROUP, LINE, PU
dependency: NODETYPE; APPN only

specifies for a PU 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.

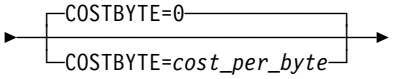
When you code CONNTYPE=APPN, XID rules for parallel TGs are enforced. So, if you have PUs with duplicate CPNAMEs, you should make them unique before this node becomes an APPN node.

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.

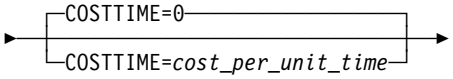
COSTBYTE



statements: GROUP, LINE, PU
dependency: 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.

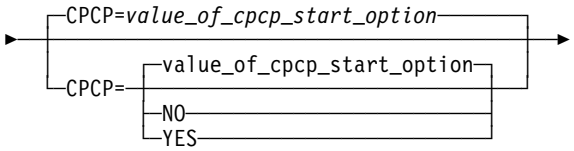
COSTTIME



statements: GROUP, LINE, PU
dependency: 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



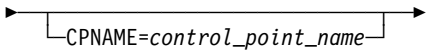
statements: GROUP, LINE, PU
dependency: APPN only

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.

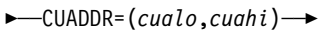
CPNAME



statements: PU
dependency: 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 in the form *netid.control_point_name*.

CUADDR



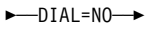
statements: PORT
dependency: none

specifies a required channel unit address pair that will be used for an X.25 port. The low address of the address pair must be specified first.

For information about address pairs in various processors, refer to the relevant processor publications.

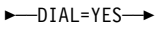
DIAL

For nonswitched lines:



statements: GROUP
dependency: applies to nonswitched lines

For switched lines:



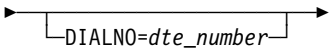
statements: GROUP
dependency: applies to switched lines

specifies whether the lines in a group require switched line control protocols.

DIAL=NO
specifies that the lines in the group represent permanent virtual circuits.

DIAL=YES
specifies that the lines in the group require switched line control protocols.

DIALNO

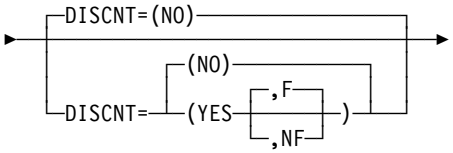


statements: PORT
dependency: none

specifies the data terminal equipment (DTE) address of this port. This number must match the number assigned by the network common carrier during network subscription. It can be 1–15 decimal digits.

Note: When calling out from the X.25 port, DIALNO must be specified for some networks; it is optional for others. Contact your network common carrier for more details.

DISCNT



statements: GROUP, LINE, PU
dependency: applies only to PU types 1, 2

specifies when VTAM should terminate 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
specifies that VTAM terminates the SSCP-LU and SSCP-PU sessions when certain conditions are met. Refer to the *VTAM Network Implementation Guide* for a discussion of those conditions.

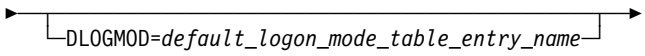
DISCNT=YES
specifies that VTAM automatically terminates the SSCP-LU and SSCP-PU sessions as soon as the last LU-LU session is terminated. Before the physical unit and its logical units can be used again, they must be reactivated by the domain operator.

DISCNT=YES,F
specifies that VTAM indicates “final-use” status in the DACTPU request unit when it deactivates a physical unit as a result of DISCNT=YES, and the physical connection can be broken. This value does not apply when DISCNT=NO, nor does it have any effect on the VARY ACT command.

DISCNT=YES,NF
specifies that “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 F or NF should be specified for a given device, consult the appropriate installation publication for the device. It also does not have any effect on the VARY INACT command.

DLOGMOD

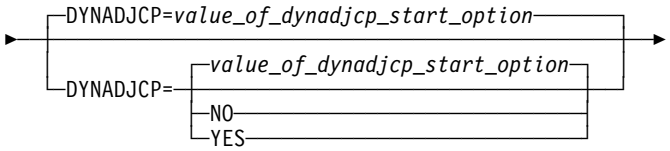


statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2

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 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 666.

DYNADJCP



statements: GROUP, LINE, PU
dependency: APPN only; ignored if CPNAME coded

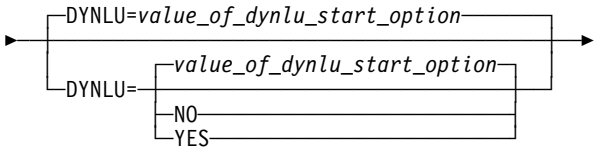
specifies whether an ADJCP can be created dynamically for an adjacent node.

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



statements: GROUP, LINE, PU
dependency: applies only to PU types 1, 2

specifies whether dynamic allocation of CDRSC definitions is 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.

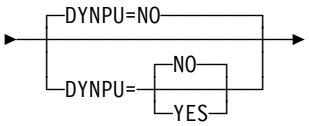
For subarea nodes (HOSTSA is specified), DYNLU=YES is meaningful only when the host CDRM is activated and CDRDYN=YES is specified on the CDRM definition statement.

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

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.

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 the description of the DYNLU operand on the ADJCP definition statement on page 23.

DYNPU



statements: GROUP (switched)
dependency: 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.

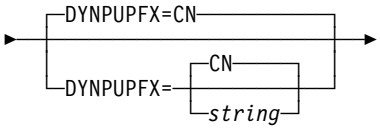
A PU created by the DYNPU operand will use the switched major node PU operand defaults, except for the following operands which will use the values noted:

- 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



statements: GROUP (switched)
dependency: 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.

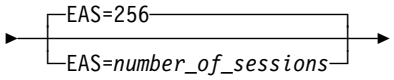
Note: This operand is valid only if DYNPU is specified as YES.

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 "Format of Definition Statements" on page 10 for naming conventions.

EAS



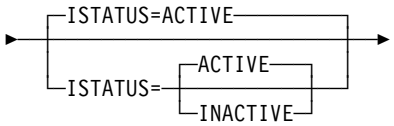
statements: GROUP, LINE, PU, LU
dependency: applies only to independent LUs (LOCADDR=0)
range: 0–65535.

specifies the estimated number of sessions that will be active with this logical unit at any given time. If you code EAS for dependent LUs, VTAM ignores it.

If your estimated value is greater than 256, for better performance specify 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 534 for information about BSBUF.

ISTATUS

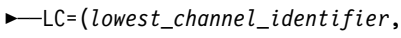


statements: GROUP, LINE, PU, LU
dependency: none

specifies whether this resource is to be activated after the first activation of the packet major node.

If you do not code the ISTATUS operand or if you code ISTATUS=ACTIVE, the resource becomes active when its major node becomes active. If you code 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. See *VTAM Operation* for information on using the VARY ACT command.

LC



statements: VCPARMS
dependency: none
range: X'000'–X'FFF'.

specifies the logical channel identifier or the range of logical channel identifiers for which this VCPARMS definition statement will apply.

lowest_channel_identifier
specifies the lowest logical channel identifier in the range of channels to which this VCPARMS definition statement applies. Valid specifications consist of one to three hexadecimal digits in the range.

highest_channel_identifier
specifies the highest logical channel identifier in the range of channels to which this VCPARMS definition statement applies. Valid specifications consist of one to three hexadecimal digits in the range.

For example, if the channel identifiers are X'002' and X'005', you code LC=(002,005).

If this VCPARMS definition statement applies to only one channel, specify that logical channel identifier for both *lcn* and *hcn*.

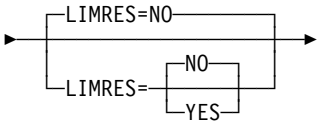
The values for logical channel identifiers must be in ascending order. Additionally, a logical channel identifier can occur only once in a group of VCPARMS definition statements.

When logical channel identifiers are for switched virtual circuits, the range of *lcn* and *hcn* must be a subset of or identical to the VCALLS specifications for *lic* and *hic*, or *ltc* and *htc*, or both.

Note: In most networks, logical channel 0 is reserved and should not be coded. Consult your network common carrier representative to learn if your network allows logical channel 0.

If this operand is coded incorrectly, this VCPARMS definition statement is ignored, and the standard default packet and window sizes are used.

LIMRES



statements: GROUP, LINE, PU
dependency: applies only to LU 6.2 conversations

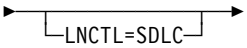
specifies whether a group of lines, a line, or a physical unit is to be treated as a limited resource. A limited resource allows you to limit the use of some network connections. It also causes any sessions that traverse the resource to be deactivated if no conversation is active. A session can also be deactivated if the time limit you specify for the LIMQSINT operand on the APPL definition statement expires.

If you do not code a value for LIMQSINT, resources are not considered as limited resources. Furthermore, sessions will remain active even after all conversations have ceased. The LIMRES operand is only applicable to PU type 1 and 2.

LIMRES=YES
specifies that this line, group of lines, or physical unit is to be treated as a limited resource.

LIMRES=NO
specifies that this line, group of lines, or physical unit is not to be treated as a limited resource.

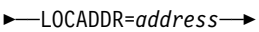
LNCTL



statements: GROUP
dependency: none

specifies that the virtual circuits under an X.25 port appear to VTAM as SDLC lines.

LOCADDR



statements: LU
dependency: see notes
range: 0–63 (PU type 1);
0–255 (PU type 2)

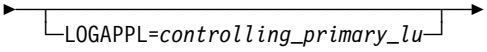
specifies the local address of the logical unit without leading zeros.

For independent logical units in a type 2.1 node, code LOCADDR=0. An independent logical unit supports multiple (including parallel) sessions and can act as a primary logical unit. 0 is not a valid value for dependent logical units. An LU definition statement is not required for every possible 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 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.

LOGAPPL



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2; NetView

indicates the name of an application program to which this logical unit is automatically logged on when the logical unit is activated. The name must correspond to the network-unique name assigned to the application program by an APPL definition statement.

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 exit. The LOGAPPL operand cannot be an independent logical unit because independent logical units cannot be the controlling logical units for automatic logon sessions.

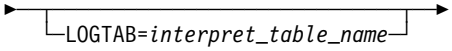
Automatic logon LU-LU sessions are not redriven for abends related to CP-CP outages even though the CP-CP sessions are redriven.

You can specify a network-qualified name for LOGAPPL. You cannot specify an LUALIAS name for LOGAPPL.

Note: If you specify the NetView program as the application program on the LOGAPPL operand, you will get unpredictable results in recovery.

For more information on establishing sessions, see the *VTAM Network Implementation Guide*.

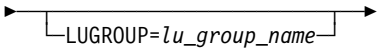
LOGTAB



statements: GROUP, LINE, PU, LU
dependency: applies only to 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 659 for a description of how to define an interpret table.

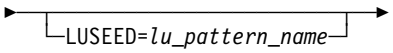
LUGROUP



statements: PU
dependency: 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



statements: PU
dependency: 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.

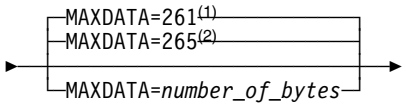
If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- 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 *VTAM Customization*.

MAXDATA



Notes:
1 MAXDATA defaults to 261 when PUTYPE=1 is coded.
2 MAXDATA defaults to 265 when PUTYPE=2 is coded.

statements: GROUP, LINE, PU
dependency: applies only to PU types 1 and 2
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/response header (RH). To determine the maximum PIU (or PIU segment) size that the physical unit can receive, consult the publications for the specific type of station represented by this PU definition statement.

The maximum amount of user data that VTAM sends to the physical unit in one PIU is:

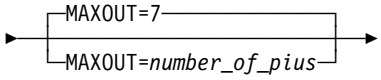
- MAXDATA minus 2 bytes (for a PU type 1)
- MAXDATA minus 6 bytes (for a PU type 2).

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 do not have to specify this value, for type 4 and 5 physical units.

If you code this operand for a type 2.1 physical unit, the value is overridden by the value taken from the format 3 XID received from the adjacent link station.

MAXOUT



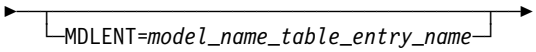
statements: PORT
dependency: none
range: 1–7

specifies the maximum number of link level I-frames that are transmitted before waiting for an acknowledgement.

For most networks, the value of MAXOUT is fixed at 7. Consult your network documentation or network common carrier representative for the value you must use.

If you code this operand for a type 2.1 physical unit, the value is overridden by the value taken from the format 3 XID received from the adjacent link station.

MDLENT

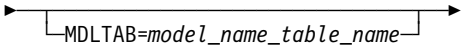


statements: GROUP, LINE, PU, LU
dependency: MDLTAB; applies only to PU types 1 and 2

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.

MDLTAB



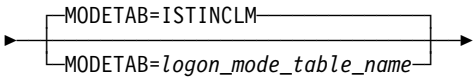
statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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. Refer to “LOGON Command” on page 714 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. See *VTAM Operation* for more information about the MODIFY TABLE command.

MODETAB



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2

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 666. If you do not supply a logon mode table for a 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

statements:	VBUILD, PORT, VCPARMS	*statements:*	GROUP, LINE, PU, LU

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 PORT: provides an optional name associated with the channel unit address pair that is configured as an X.25 port on your integrated communication adapter. Although a 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 VCPARMS: provides an optional name for the group of virtual circuit values.

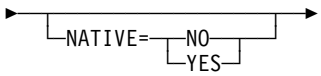
On GROUP: provides a required minor node name for the line group. It contains 1–8 alphanumeric characters beginning with an alphabetical character other than $.

On LINE: provides a required minor node name for the line. It contains 1–8 alphanumeric characters beginning with an alphabetic character other than $.

On PU: provides a required minor node name of the physical unit represented by this definition statement.

On LU: provides a required minor node name for the logical unit.

NATIVE



statements: PU
dependency: 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, and

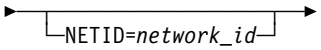
- if the NETIDs match, the connection defaults to a native connection
- if the NETIDs differ, the connection defaults to a nonnative connection.

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, the SSCP-SSCP session will also fail.

NETID



statements: PU
dependency: applies only to PU types 1 and 2

specifies a 1–8 character network name.

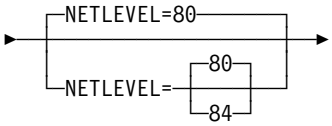
Note: The use of the national character, #, is discouraged, because this character might not be present on keyboards of terminals produced in other countries. Because # might not be available on all terminals, SNA architecture specifically excludes # from the list of valid characters that can be used for defining network names (NETIDs). Although VTAM allows you to use #, other products might enforce this restriction.

For type 2 physical units: This operand is used in conjunction with the XNETALS start option in support of nonnative network connection. See page 600 for information about XNETALS. This operand applies to type 2 physical units only.

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 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.

NETLEVEL



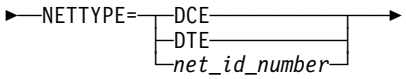
statements: PORT
dependency: none

specifies the level of CCITT standards used in the X.25 network.

NETLEVEL=80
specifies that the network is at the level of the 1980 CCITT X.25 Recommendation.

NETLEVEL=84
specifies that the network is at the level of the 1984 CCITT X.25 Recommendation.

NETTYPE



statements: PORT
dependency: none

specifies whether the X.25 port operates in a DTE-to-DTE configuration or is attached to a packet-switched data network (PSDN).

NETTYPE=DCE
specifies that the X.25 port is connected (using modems or a modem eliminator) to another DTE rather than to a PSDN. The DTE will assume the role of the data circuit-terminating equipment (DCE).

NETTYPE=DTE
specifies that the X.25 port is connected (using modems or a modem eliminator) to another DTE rather than to a PSDN. The other DTE will assume the role of the DTE.

NETTYPE=net_id_number
specifies the identification number of the PSDN that can be attached through this port. The identification number tells VTAM what differences must be accommodated for a specific PSDN.

Table 36 on page 386 lists the valid values for *net_id_number* and the represented networks.

Table 36. Network Types for VTAM X.25 Support

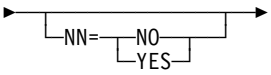
NETTYPE Value	Vendors' Network Names
1	AT&T DATAPAK DATEX-AC DATEX-P EIRPAC ENTEL ERICSON INFONET INTELPAK ISRANET MAYPAC MCI* PAXNET RC/KTAS/JT SWITCHSTREAM-1 TELENET TELEPAC TYMNET VENUS-P YUPAK
2	TRANSPAC XI
3 and 4	Reserved
5	AUSTPAC DACOM-NET DATAPAK (Finland) LUXPAC NZPO PACNET RENPAC
6	ARPAC IBERPAC INFOSWITCH
7	DCS ITAPAC TELEPAC (Singapore)
8	DDX-P SAPONET
9	DATANET-1
10 and 11	Reserved

Note: VSE The values 3, 4, 10, and 11 are reserved. If you have previously used these values to define your networks, there is no need to change them. Network types that were defined with 3, 4, 10, and 11 are now defined as shown in Table 37 on page 387.

Table 37. Valid Network Types from Previous Releases of VTAM (VSE/ESA)

Previous NETTYPE Value	Vendors' Network Names	Current NETTYPE Value
3	TELEPAC — Switzerland	1
4	DATAPAC — Denmark DATAPAK — Norway ISRANET — Israel SWITCHSTREAM1 — U.K. TELEPAC — Mexico	1
10	VENUS-P — Japan	1
11	DDX-P — Japan	8

NN



statements: GROUP, LINE, PU
dependency: APPN only

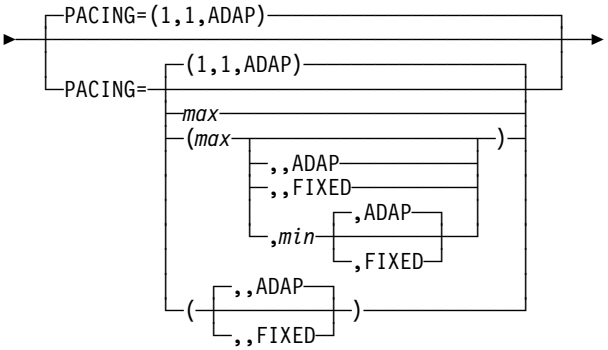
specifies whether the adjacent node is expected to be a network node.

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, connection setup fails. If NN is not specified, 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



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2
range: 1–63

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.

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 have leading zeros but is limited to eight characters.

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)

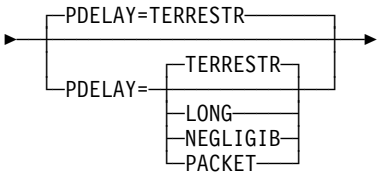
type is used only by NCP, and it requires at least NCP V5R4.

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 *VTAM Network Implementation Guide*.

PDELAY



statements: GROUP, LINE, PU
dependency: 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.

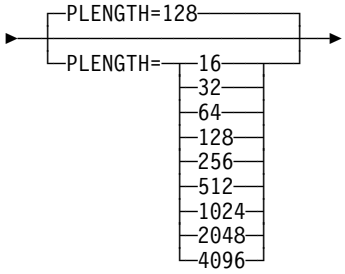
PDELAY=LONG
indicates a satellite delay (greater than 245.76 milliseconds).

PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

PDELAY=PACKET
indicates a packet-switched network delay (between 49.152 and 245.76 milliseconds).

PDELAY=TERRESTR
indicates telephone network delay (between .48 and 49.152 milliseconds).

PLENGTH



statements: PORT, VCPARMS
dependency: see description

On PORT: specifies the maximum length of the user data field selected for nonstandard default packet size. This packet size is common to all switched virtual circuits defined by the VCALLS operand. The size is the same for both directions of transmission. To specify a packet size for switched virtual circuits that is different from the default, you must have subscribed to the optional user facility, Nonstandard Default Packet Size. To specify a packet size for permanent virtual circuits that is different from the default, see “VCPARMS” on page 363.

Notes:

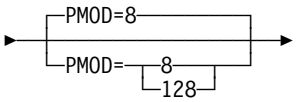
1. If your packet switched data network (PSDN) uses a packet size other than 128 bytes, or if you subscribe to an extended facility that uses a packet size other than 128 bytes, you must specify a value for PLENGTH that matches that packet size. You must not allow PLENGTH to default to 128 bytes.
2. When NETTYPE=DTE or NETTYPE=DCE, this PLENGTH value must match the nonstandard default packet size defined for the same logical channels of the remote DTE.

On VCPARMS: specifies the maximum length of the user data field in a data packet. The value is the same for both directions of transmission.

For permanent virtual circuits, the value specified must be agreed upon by your installation and the packet-switched data network (PSDN). For switched virtual circuits, the PLENGTH value specifies the limit to which the packet size is negotiated.

The value specified on the PLENGTH operand of the PORT definition statement is always used when the call originates from your host.

PMOD



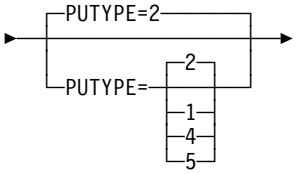
statements: PORT
dependency: NETTYPE

specifies the packet sequence number modulo.

If you specify 128, you are specifying Extended Packet Sequence Numbering, an optional user facility allowed by some networks. To specify PMOD=128, you must have specified this optional user facility when you subscribed to the packet-switched data network.

For ES/9370, 9221, and 9371 processors, when NETTYPE=DTE or NETTYPE=DCE, this PMOD value must match the packet sequence number modulo defined for the same logical channels of the remote DTE.

PUTYPE



statements: GROUP, LINE, PU
dependency: none

specifies the physical type of the SNA station represented by this PU definition statement:

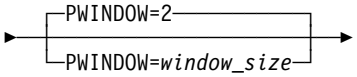
PUTYPE=1
indicates an SNA terminal.

PUTYPE=2
indicates an SNA cluster controller or a type 2.1 node.

PUTYPE=4
indicates another domain's NCP (with NPSI) in a communication controller.

PUTYPE=5
indicates another domain's VTAM and an integrated communication adapter.

PWINDOW



statements: PORT, VCPARMS
dependency: see description
range: 1 ≤ PWINDOW value < PMOD value

On PORT: specifies the window size selected as the nonstandard default window size common to all switched virtual circuits defined by the VCALLS operand.

To specify a window size for switched virtual circuits that is different from the

default, you must have subscribed to the optional user facility, Nonstandard Default Window Size. To specify a window size for permanent virtual circuits that is different from the default, see “VCPARMS” on page 363.

Notes:

1. The window size is the same for both directions of the transmission. If the window size for both directions of the DTE/DCE connection is not the same, hung sessions occur.
2. If your PSDN uses a window size other than 2, or if you subscribe to an extended facility that uses a window size other than 2, you must specify a value for PWINDOW that matches that window size. You must not allow PWINDOW to default to 2.
3. For ES/9370, 9221, and 9371 processors, when NETTYPE=DTE or NETTYPE=DCE, this PWINDOW value must match the nonstandard default window size defined for the same logical channels of the remote DTE.

On VCPARMS: specifies the window size used for each logical channel defined on this VCPARMS definition statement.

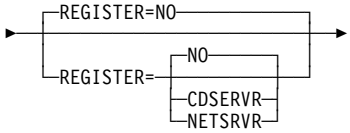
Note: The window size is the same for both directions of the transmission. If the window sizes are not the same, hung sessions will occur.

For permanent virtual circuits, the value specified must be agreed upon by your installation and by the PSDN. For switched virtual circuits, the PWINDOW value specifies the limit to which the window size is negotiated.

The value specified on the PWINDOW operand of the PORT definition statement is always used when the call originates from your host.

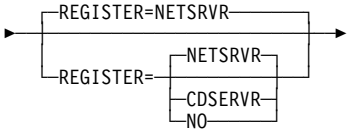
REGISTER

For Independent LUs:



statements: PU, LU
dependency: APPN only

For Dependent LUs:



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.

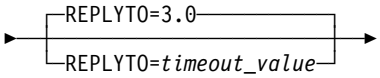
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

REPLYTO



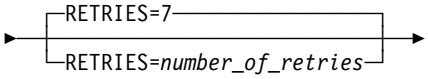
statements: PORT
dependency: none
range: 0.1–25.5 seconds

specifies, in seconds and tenths of seconds, the time allowed by the DTE between the transmission of frames and receipt of an acknowledgement frame. The value of REPLYTO must match the actual speed of the physical access line.

When the port is to be used in conjunction with a PSDN, check with your network common carrier for the values allowed or required by your network.

For ES/9370, 9221, and 9371 processors, for the DTE-to-DTE mode of operation, the value you specify for REPLYTO should be based on the characteristics of the equipment connecting the two DTEs. If you specify a value larger than necessary, VTAM might have unexpected delays in determining that the partner DTE has become inoperative.

RETRIES

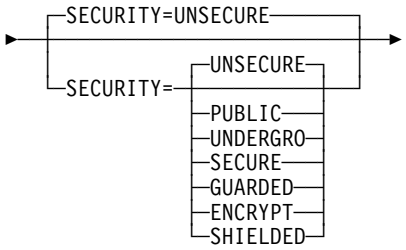


statements: PORT
dependency: none
range: 3–31

specifies the maximum number of times a frame is retransmitted following expiration of the REPLYTO timer interval.

For ES/9370, 9221, and 9371 processors, for the DTE-to-DTE mode of operation, the value you code for REPLYTO should be based on the characteristics of the equipment connecting the two DTEs. If you specify a value larger than necessary, VTAM might have unexpected delays in determining that the partner DTE has become inoperative.

SECURITY (VM)



statements: GROUP, LINE, PU
dependency: 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.

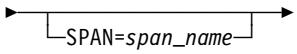
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SPAN

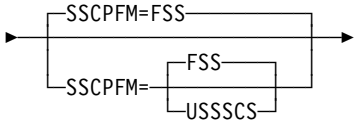


statements: GROUP, LINE, PU, LU
dependency: NetView

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

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

SSCPFM



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 2

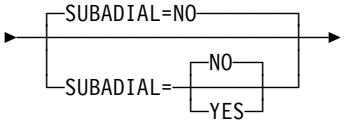
determines what type of RUs the logical unit can support in its communications with the SSCP.

SSCPFM=FSS
specifies that the logical unit supports formatted commands in its communications with the SSCP.

SSCPFM=USSSCS
specifies that the logical unit supports character-coded commands in its communications with the SSCP.

See the publications for each individual device to determine whether formatted or character-coded commands are supported by the device.

SUBADIAL



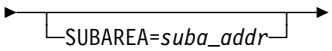
statements: GROUP
dependency: none

specifies whether the lines in the group are used for switched subarea connections.

SUBADIAL=NO
specifies that the lines in the group are not switched subarea lines.

SUBADIAL=YES
specifies that the lines in the group are switched subarea lines.

SUBAREA

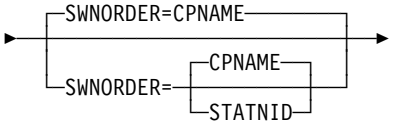


statements: PU
dependency: applies only to subarea nodes (PUTYPE=4 or 5)
range: 1—value of MXSUBNUM start option

specifies the subarea address assigned to the physical unit (NCP or VTAM) represented by this PU definition statement. VTAM uses the subarea address to determine which of two physical units is primary; the one with the highest subarea is always primary.

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.

SWNORDER



statements: GROUP, LINE
dependency: none

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).

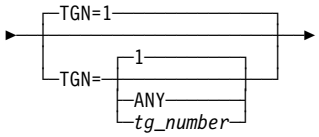
This operand overrides the SWNORDER start option value.

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.

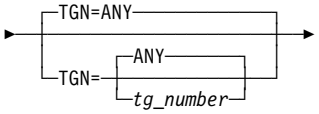
TGN

For subarea TGN:



statements:	PU
dependency:	PUTYPE=4,5
range:	1–255

For APPN TGN:



statements:	PU
dependency:	PUTYPE=2, CPNAME, NETID
range:	0–21

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 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, a value for CPNAME and NETID must also be specified.

TGN=*tg_number*
specifies a decimal integer assigned to the transmission group.

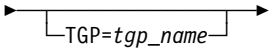
For APPN, coding TGN=0 means that the PU does not support parallel TGs.

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. 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

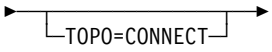


statements: GROUP, LINE, PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO



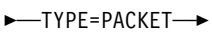
statements: PU
dependency: APPN only

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.

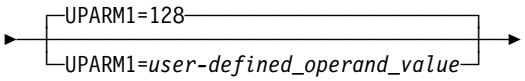
TYPE



statements: VBUILD
dependency: none

specifies that this VBUILD definition statement defines a packet major node to VTAM.

UPARM1

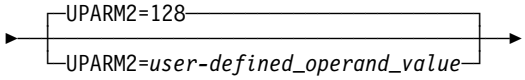


statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, "Forcing an APPN Route in a VTAM Network" in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

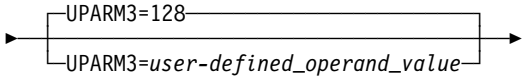
UPARM2



statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

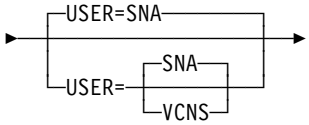
UPARM3



statements: GROUP, LINE, PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

USER



statements: LINE
dependency: see USER=VCNS description

specifies whether the line supports SNA connections or VTAM Common Network Services (VCNS) connections.

USER=SNA
specifies that the line supports SNA connections.

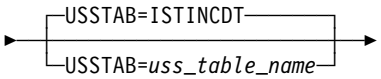
USER=VCNS
VM specifies that the line supports VCNS connections. An X.25 port cannot be used for VCNS-based communication until a line defined with USER=VCNS is activated.

The following restrictions apply to a line defined as USER=VCNS:

- The line must reside in a nonswitched line group. In other words, you must specify DIAL=NO on the GROUP definition statement.
- There must be no physical unit or logical unit resources subordinate to the line. In other words, no PU or LU definition statements can follow the LINE definition statement.
- You can code only one LINE definition statement specifying USER=VCNS within the port definition.

When USER=VCNS on the LINE definition statement, ISTATUS is the only other operand that applies.

USSTAB

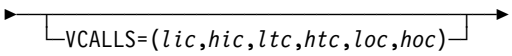


statements: GROUP, LINE, PU, LU
dependency: applies only to dependent LUs

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 690.

VCALLS



statements: PORT
dependency: required for switched virtual circuits; NETTYPE
range: X'000'–X'FFF', for each specification

defines how the switched virtual circuits are allocated among the logical channels on this port.

lic,hic
specifies the range of logical channel identifiers for one-way logical channels incoming. This combination is required when you subscribe to one-way logical channels incoming.

lic specifies the identifier of the lowest incoming channel. *hic* specifies the identifier of the highest incoming channel.

Note: Any logical channel identifiers for permanent virtual circuits must be lower than this range.

ltc,htc
specifies the range of logical channel identifiers for two-way channels.

ltc specifies the identifier of the lowest two-way channel. *htc* specifies the identifier of the highest two-way channel.

loc,hoc
specifies the range of logical channel identifiers for one-way channels outgoing. This combination is required when you subscribe to one-way logical channels outgoing.

loc specifies the identifier of the lowest outgoing channel. *hoc* specifies the identifier of the highest outgoing channel.

Note: Although you can specify the value 000, most packet-switched data networks reserve use of logical channel 0. If your network reserves use of logical channel 0, do not specify 000.

The range of value pairs must be in increasing number order, such that,

lic is less than or the same as *hic*.

ltc is greater than *hic* but less than or the same as *htc*.

loc is greater than *htc* but less than or the same as *hoc*.

If you omit a specification, commas must be inserted with no blank spaces unless they trail the last specification. For example, to specify two-way logical channels and one-way logical channels outgoing but no one-way logical channels incoming, you might use the following:

```
VCALLS=(,,004,005,006,008)
```

To specify one-way logical channels incoming and outgoing but no two-way logical channels, you might use the following:

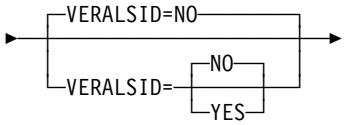
```
VCALLS=(004,005,,,006,008)
```

To specify one-way logical channels incoming and two-way logical channels but no one-way logical channels outgoing, you might use the following:

```
VCALLS=(004,005,006,008)
```

For ES/9370, 9221, and 9371 processors, when NETTYPE=DTE, code VCALLS to define the switched virtual circuits from the perspective of the remote DTE. By coding the same values for both VTAMs in a DTE-DTE configuration, you ensure that the switched virtual circuits are identical.

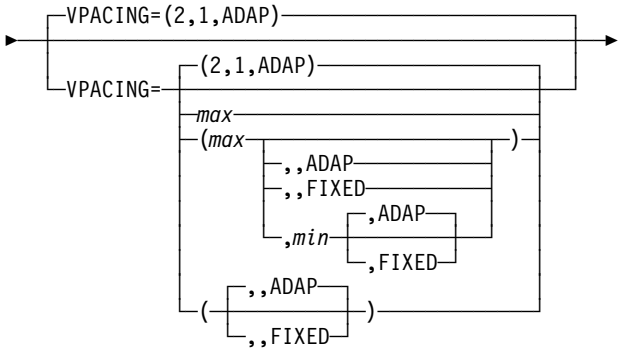
VERALSID



statements: GROUP, LINE, PU
dependency: 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



statements: GROUP, LINE, PU, LU
dependency: applies only to PU types 1 and 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.

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).

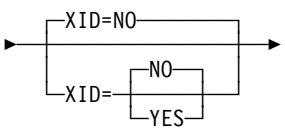
  type is used only by NCP, and it requires at least NCP V5R4.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

XID



statements: GROUP, LINE, PU
dependency: used primarily with type 2.1 peripheral nodes

specifies how the physical unit is polled.

XID=NO
specifies that the physical unit is contact polled with a Set Normal Response Mode (SNRM) command. If you specify NO, or use it by default, an XID is never sent to the physical unit. For type 1 and 2 physical units, specify XID=NO.

XID=YES
specifies that polling is done by sending an initial XID, (a null XID). Specify YES for any physical unit that supports XID in normal disconnected response mode. For a type 2.1 physical unit, specify XID=YES.

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:

- Channel-attachment **VM,VSE**
- External communication adapter (XCA)
- LAN **VM,VSE**
- NCP
- Packet **VM,VSE**

These switched connections can be made to a communication controller that is attached to a host processor. **VM** The switched connections can be made to a type 2.1 peripheral node. These connections can also be made to a host through an IBM 3172 Interconnect Controller.

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.

These lines are defined in switched line groups in a channel-attachment major node (VM, VSE) or an NCP major node.

For dial-out operations, the PATH definition statement defines the paths used to connect the communication controller or host processor and the other subarea.

Refer to “Switched SDLC Subarea Connection” in the *VTAM Network Implementation Guide* for examples and additional information on switched subareas.

PU, PATH, and LU Definition Statements for X.25 (VM, VSE)

The PU and LU definition statements define physical units and logical units on switched lines that correspond to switched virtual circuits through an X.25 port. These lines are defined in switched line groups in the packet major node.

For call-out operations, the PATH definition statement defines paths used to connect an X.25 port and the physical unit.

Refer to “X.25 Connection” in the *VTAM Network Implementation Guide* for examples and additional information on X.25.

PU, PATH, and LU Definition Statements for the LAN Major Node (VM, VSE)

The PU and LU definition statements define peripheral physical units and logical units on switched lines that can be dialed in to or dialed out from a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem. These lines are defined in switched line groups in the LAN major node. For dial-out operations, the PATH definition statement defines the paths used to connect the physical unit and a processor with its IBM Token-Ring Subsystem Controller or a processor with its multi-protocol communication subsystem.

If you have a switched major node defined in your network, you can use a copy of that definition for the switched major node in the LAN environment, and add the operands required for the LAN environment. Operands coded in your present definition that are not required for the LAN environment are allowed, but will have no meaning.

Refer to "External Communication Adapter (XCA) Connections" in the *VTAM Network Implementation Guide* for examples and additional information on local area networks.

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 Interconnect Controller. 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 Interconnect Controller.

Table 38 (Page 1 of 4). Switched Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=SWNET	455	R			
		CONFGDS	415				MVS,VSE
		CONFGPW	416				MVS,VSE
		MAXDLUR	440				CONFGDS NN and ICN only; DLURNAME
		MAXGRP	441				PATH
		MAXNO	441				PATH
name	PU	ADDR	412	R			nonswitched connections
		ANS	412			STOP	
		ASDP	413			NO	
		AUTHLEN	414			YES	PU Type 2.1; APPN only
		CAPACITY	415			8K	APPN only
		CONNTYPE	416			CONNTYPE start option	APPN only; NODETYPE
		COSTBYTE	417			0	APPN only
		COSTTIME	417			0	APPN only; TOPO

Table 38 (Page 2 of 4). Switched Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		CPCP	417			CPCP start option	APPN only
		CPNAME	417				PU Type 2.1
		DATMODE	418			HALF	
		DISCNT	421			(NO)	
		DWACT	425			NO	
		DYNADJCP	426			DYNADJCP start option	APPN only; CPNAME
		DYNLU	426			DYNLU start option	PUTYPE=2; CDRDYN
		IDBLK	430				CPNAME; IDNUM
		IDNUM	431				CPNAME; IDBLK; PUTYPE=4,5
		IRETRY	432			NO	PUTYPE=1,2
		ISTATUS	433			ACTIVE	
		LANACK	433			(0,0)	VM,VSE see description
		LANCON	433				VM,VSE see description
		LANINACT	434			4.8	VM,VSE see description
		LANRESP	434				VM,VSE see description
		LANSDWDW	435			(2,1)	VM,VSE see description
		LANSW	435			NO	VM,VSE see description
		LIMRES	436			NO	LIMQSINT; LU 6.2; PUTYPE=1, 2
		LUGROUP	438				
		LUSEED	439				LUGROUP
		MACADDR	439				VM,VSE ; LAN
		MAXDATA	440			261 (PU Type 1) 265 (PU Type 2)	see description
		MAXOUT	441			1	3172 controller
		MAXPATH	441			0	
		NATIVE	443				BN=YES APPN only
		NETID	444				XNETALS start option
		NN	445				APPN only
		PASSLIM	446			1	PUTYPE=1,2
		PDELAY	446			TERRESTR	APPN only
		PRTCT	447				PUTYPE=4,5
		PUTYPE	447			2	
		SAPADDR	449			4	VM,VSE LAN
		SECNET	449			NO	PUTYPE=1,2
		SECURITY	450			UNSECURE	APPN only MVS,VM

Table 38 (Page 3 of 4). Switched Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		SPAN	451				NetView
		SUBAREA	452			*in_xid_value*	PUTYPE=4,5
		TGN	453				see description
		TGP	454				APPN only
		TOPO	454			CONNECT	APPN only
		UPARM1	455			128	APPN only
		UPARM2	455			128	APPN only
		UPARM3	455			128	APPN only
		VERALSID	456			NO	APPN only
name	PATH	CALL	414			INOUT	GRPNM
		DIALNO	418				LINENM
		DLCADDR	422				LINENM
		DLURNAME	425				DLCADDR
		GID	430				
		GRPNM	430				CALL VERID VERIFY
		LINENM	437				DIALNO DLCADDR
		PID	447				
		REDIAL	448			3	
		SHM	450			NO	DIALNO GRPNM
		SHMTIM	451			0.0	
		SHOLD	451				VSE
		USE	455			YES	
		VERID	457				GRPNM
		VERIFY	457			NONE	GRPNM; VERID
name	LU	LOCADDR	437	R			
		ASLENT	413		P		ASLTAB PUTYPE=1,2
		ASLTAB	413		P		PUTYPE=1,2
		BATCH	414		P	NO	pre-V4R3 NCPs PUTYPE=1,2
		DLOGMOD	424		P		PUTYPE=1,2
		EAS	427		P	256	VM,VSE see description
		ENCR	427		P	NONE	MVS
		FASTPASS	428		P	YES	TSO only
		FEATUR2	428		P	(DUALCSE, MODEL1, NOEDATS, NOPRINTR, NOSELPEN)	
		ISTATUS	433			ACTIVE	
		LOGAPPL	437		P		PUTYPE=1,2
		LOGTAB	438		P		PUTYPE=1,2
		MAXSESS	442		P		NCP-attached independent LUs
		MDLENT	442		P		MDLTAB PUTYPE=1,2

Table 38 (Page 4 of 4). Switched Major Node Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		MDLTAB	442		P		PUTYPE=1,2
		MODETAB	443		P	ISTINCLM	PUTYPE=1,2
		PACING	445		P	1	PUTYPE=1,2
		REGISTER	448		P	see description	APPN only
		RESSCB	449			0	independent LU
		SPAN	451				NetView
		SSCPFM	452		P	FSS	PUTYPE=1,2
		TERM	453		P		PUTYPE=1,2
		USSTAB	456		P	ISTINCDT	PUTYPE=1,2; dependent LUs
		VPACING	458		P	2	PUTYPE=1,2

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 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.

A PATH definition statement is required if short-hold mode, 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
CAPACITY
CONNTYPE
COSTBYTE
COSTTIME
CPCP
DATMODE
DISCNT
DLOGMOD
DYNADJCP
DYNLU
ENCR
FASTPASS
FEATUR2
IDBLK
IDNUM
IRETRY
ISTATUS
LANACK
LANCON
LANINACT
LANRESP
LANSDWDW
LANSW
LIMRES
LOCADDR
LOGAPPL
LOGTAB
LUGROUP
LUSEED
MACADDR
MAXDATA
MAXOUT
MAXSESS
MDLENT
MDLTAB
MODETAB
NN
PACING
PASSLIM
PDELAY
REGISTER
SAPADDR
SECURITY
SECNET
SSCPFM
TERM
TGP
TOPO
UPARM1
UPARM2
UPARM3
USSTAB
VERALSID
VPACING

To dynamically change the operand, 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 tables (ASLTAB, LOGTAB, MDLTAB, MODETAB, and USSTAB) at any time without inactivating resources.

Refer to “VARY ACT Command” in *VTAM Operation* and “Dynamic Reconfiguration and Change of Operands” in the *VTAM 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.

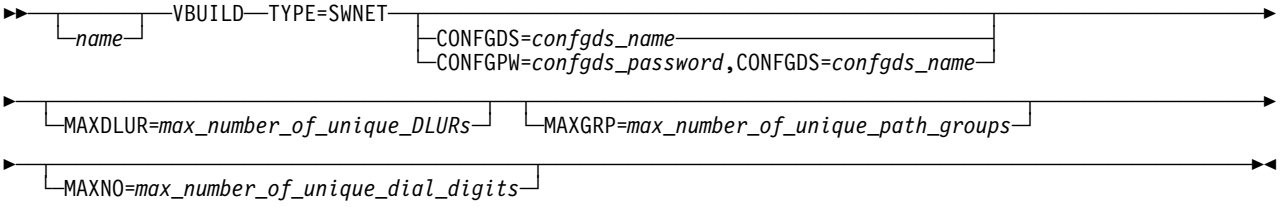
The NODETYPE start option allows APPN function. The combination of

NODETYPE and the HOSTSA start option determines the various configurations of subarea node, interchange node, migration data host, network node, or end node.

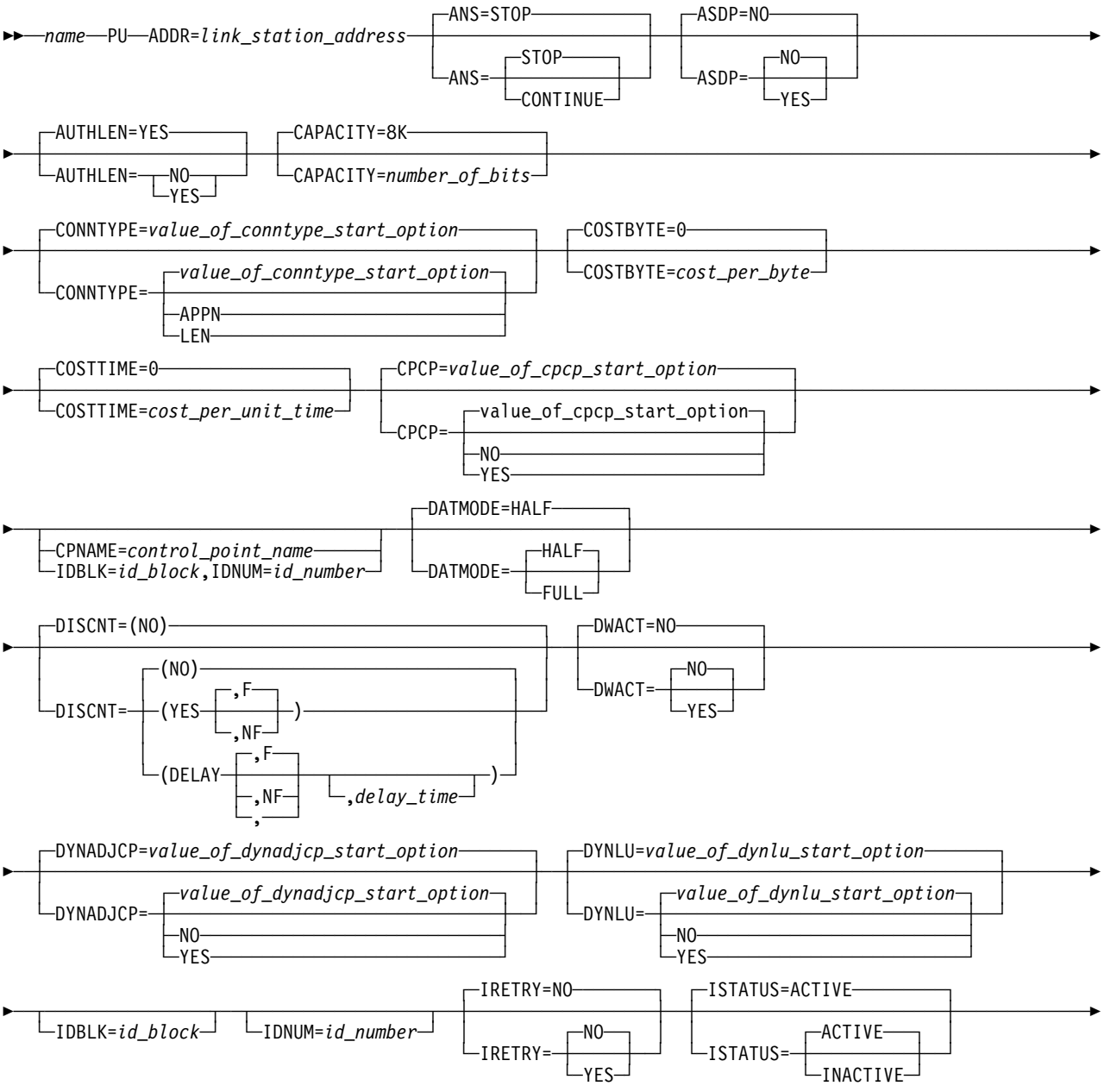
Full Syntax

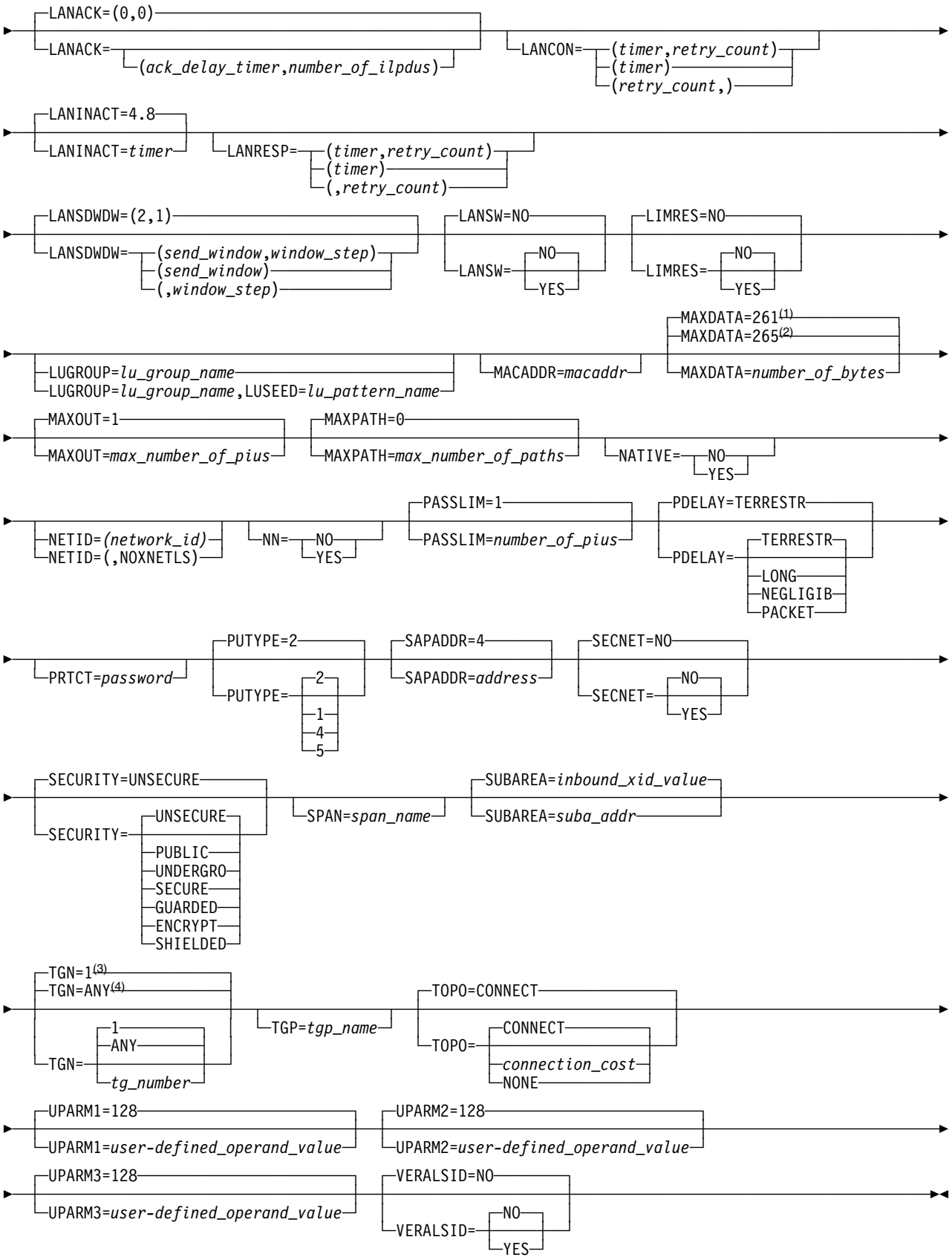
The full syntax for the switched major node follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

VBUILD



PU





Notes:

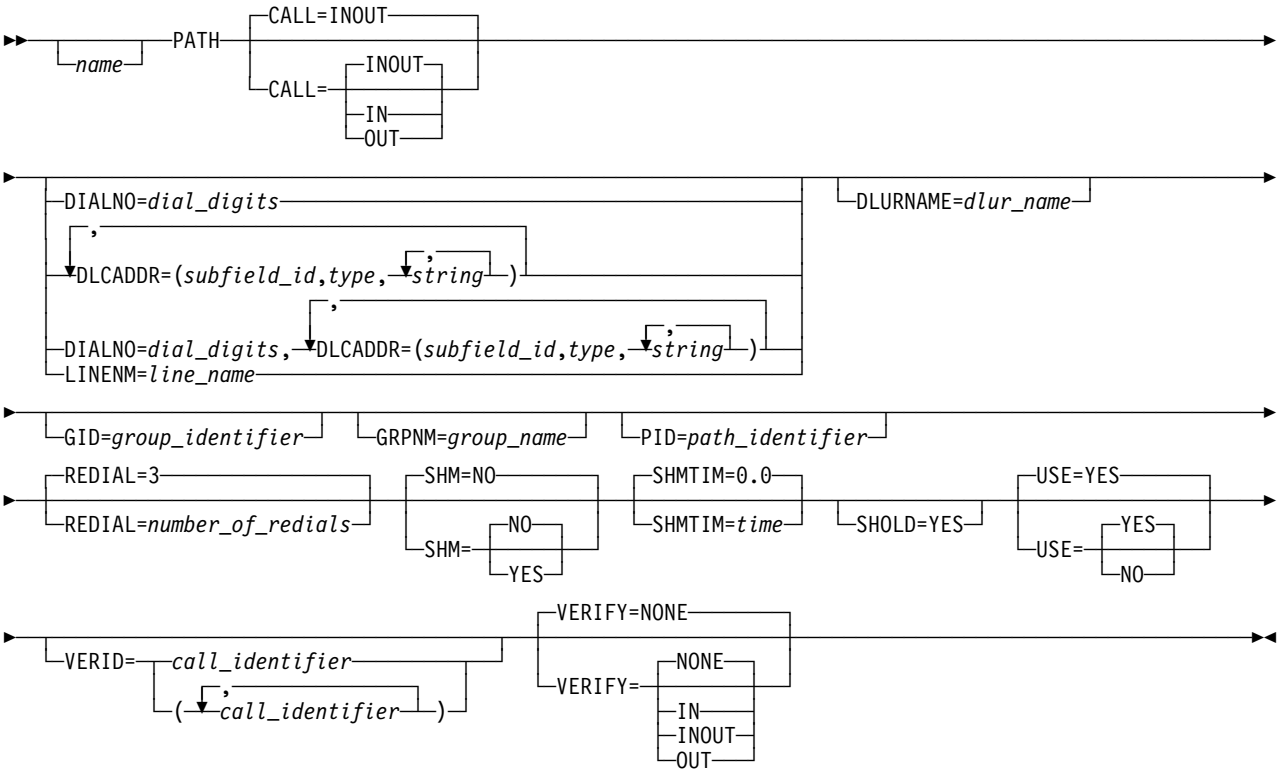
1 MAXDATA defaults to 261 when PUTYPE=1 is coded.

2 MAXDATA defaults to 265 when PUTYPE=2 is coded.

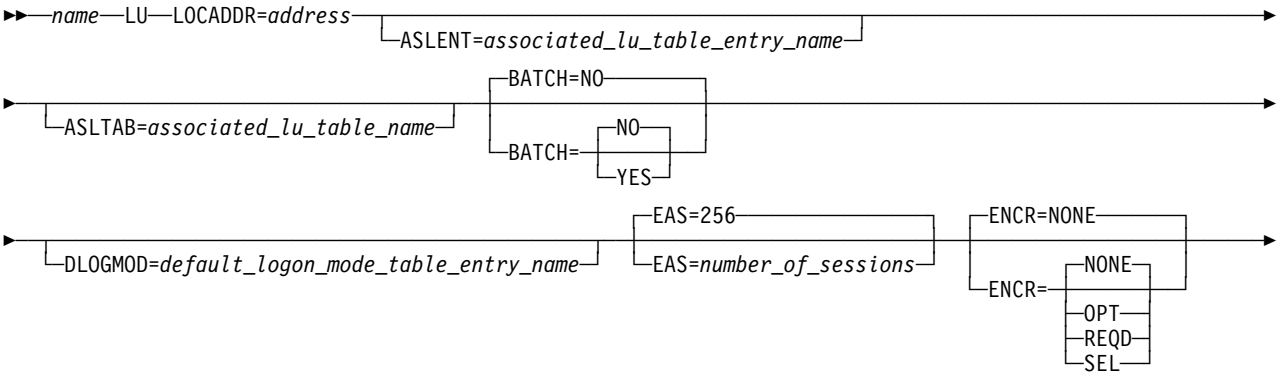
[3] TGN defaults to 1 for subarea.

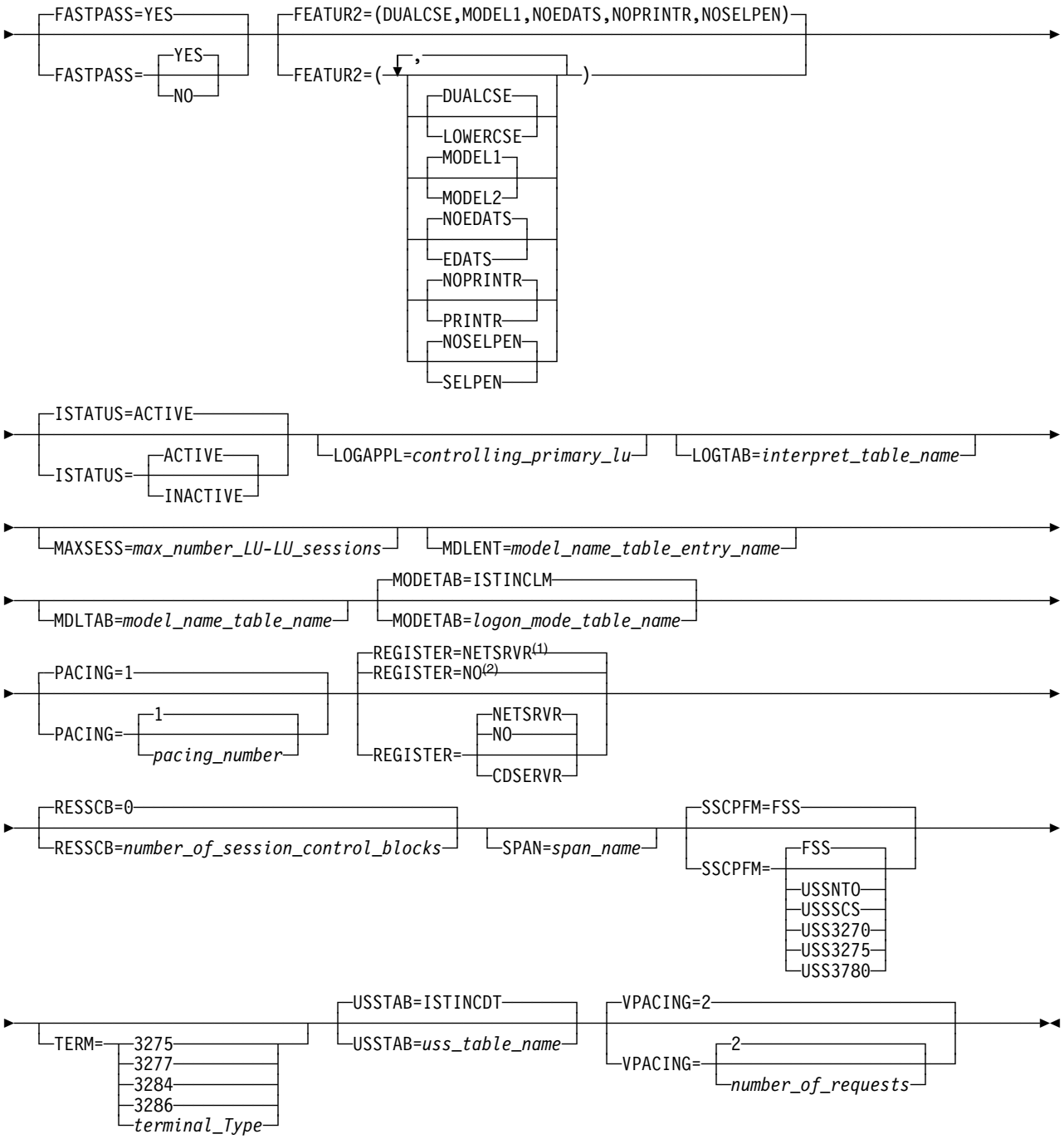
[4] TGN defaults to ANY for APPN.

PATH



LU





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 switched major node.

ADDR

statements: PU
dependency: nonswitched connections

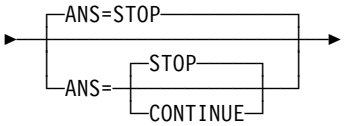
specifies the hexadecimal SDLC station address for the physical unit. You can specify one or two hexadecimal digits for ADDR. For example, if the address is X'03', you can code ADDR=03 or ADDR=3.

This address is required for all physical units attached through SDLC lines and X.25 networks.

This operand is ignored for switched subarea connections.

VM If you are using an ES/9370 processor with a 37x5 running NCP/Token-Ring interconnection, you can omit this operand. However, note that a warning message will be issued.

ANS



statements: PU
dependency: none

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.

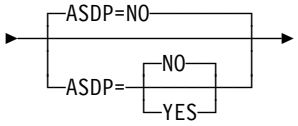
MVS Code ANS=CONTINUE if the terminal participates in XRF sessions.

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 “SSCP Takeover” in the *VTAM Network Implementation Guide*.

ANS=STOP
indicates that the connection should be dropped.

ASDP



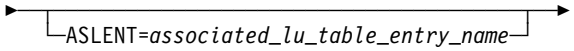
statements: PU
dependency: none

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

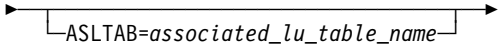


statements: PU, LU
dependency: ASLTAB; PUTYPE=1,2

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



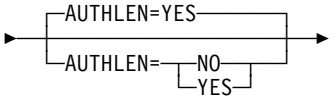
statements: PU, LU
dependency: PUTYPE=1 or 2

specifies the name of the associated LU table to be used for the logical unit. See “Associated LU Table” on page 638 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

AUTHLEN



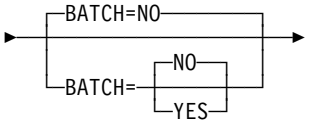
statements: PU
dependency: PU Type 2.1; APPN only

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



statements: PU, LU
dependency: PUTYPE=1 or 2; pre-V4R3 NCPs only

specifies the processing priority that the NCP uses for the logical unit.

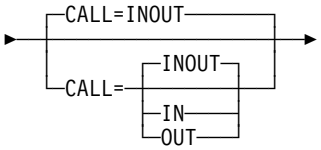
BATCH=NO
BATCH=NO means a high priority (suitable for interactive application programs).

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



statements: PATH
dependency: 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.

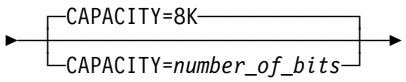
CALL=IN
specifies that inbound calls can be accepted but outbound calls cannot be made using this PATH definition statement.

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

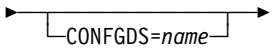


statements: PU
dependency: APPN only

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).

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 50 on page 497 for a list of CAPACITY values you can specify and their corresponding values when displayed.

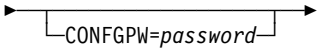
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

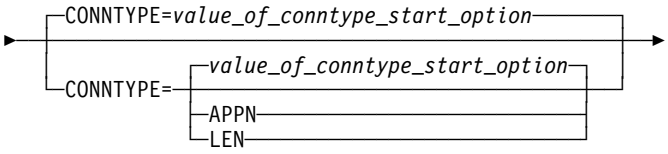


statements: VBUILD
dependency: CONFGDS

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



statements: PU
dependency: 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.

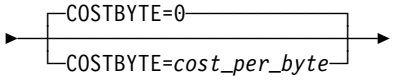
CONNTYPE=LEN
specifies that this connection does not support parallel TGs, CP-CP sessions, or CP name change support.

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.

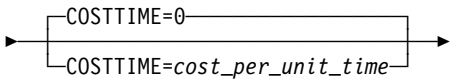
COSTBYTE



statements: PU
dependency: 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

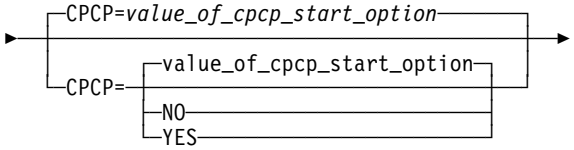


statements: PU
dependency: APPN only; TOPO
range: 0–255

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 via 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



statements: PU
dependency: APPN only

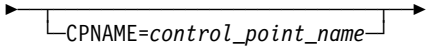
specifies whether CP-CP sessions are supported on this connection. If you do not code CPCP, the value defaults to YES only if you specify YES on the CPCP start option. Otherwise, the value defaults to NO.

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



statements: PU
dependency: PU Type 2.1

specifies the control point (CP) name of a type 2.1 peripheral node. CPNAME is recommended for type 2.1 peripheral nodes that support it.

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.

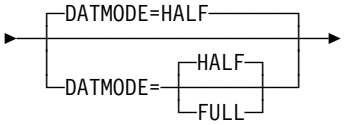
VTAM uses the CP name of a type 2.1 PU to identify the PU in any messages. *control_point_name* must differ from the PU *name*, or a sense code is given.

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. Subsequent PUs supported by the DLUR require a different means of identification dependent upon DLC support (for example, IDBLK/IDNUM for Internal PUs). Do not code CPNAME when the DLUR is not adjacent (adjacent means having CP-CP sessions with the DLUR). Additional information regarding the CPNAME parameter can be found in the DLUR publications.

DATMODE



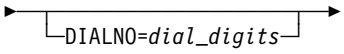
statements: PU
dependency: none

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



statements: PATH
dependency: LINENM

specifies the dial information used to initiate a connection with a physical unit over a switched link. **VM, VSE** DIALNO must be specified for switched lines that can

call out to a remote physical unit (that is, for lines specified with CALL=OUT or CALL=INOUT on the LINE definition statement under a packet major node).

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.

Note: VM For VM and a 9371 processor, you can use this operand to pass V.25 bis commands to a modem that can handle V.25 bis commands.

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=aabbccccccccccc
```

where:

aa is the token-ring interface coupler (TIC) number of the communication controller (in the range 00–99). The TIC number equates with the PORTADD keyword on the NCP LINE definition statement.

bb is the service access point (SAP) address of the terminal. It must be a multiple of 4. See "SAPADDR (VM, VSE)" on page 449 for more information.

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.

aa and *bb* are decimal numbers.

Note: The last 12 digits of DIALNO are equivalent to the twelve digits of the MACADDR operand on the PU definition statement. 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.

IBM 3172 Interconnect Controller Switched Data Networks

Use the following DIALNO format for host-initiated connections to peripheral devices connected through the IBM 3172 Interconnect Controller (*aa*, *bb*, and *cccccccccccc* are hexadecimal):

```
DIALNO=aabbcccccccccccc
```

where:

aa is a 2 digit place-holder. It has no meaning in the IBM 3172 Interconnect Controller environment.

bb is the service access point (SAP) address of the terminal. It must be a multiple of 4. VM See "SAPADDR (VM, VSE)" on page 449 for more information.

cccccccccccc is 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 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.

X.25 Packet-Switched Data Networks (VM, VSE)

DIALNO consists of the called address and optional user facility characters. The called address of the remote physical unit must be at the beginning of the DIALNO specification and can be from 1–15 characters. The called address portion of DIALNO is required. The address length is network-specific; it depends on the requirements of each packet-switched data network. Consult your network subscription or your network common carrier for information on the called address.

Optional user facility characters indicate certain optional user facilities. They are appended after the called address in any order.

- RPOA identifier: If the remote physical unit is in a different packet-switched data network and the call requires a recognized private operating agency (RPOA), include the RPOA identifier in the DIALNO. The letter “P” precedes the RPOA identifier.
- CUG identifier: If the remote physical unit is a member of more than one CUG, code the CUG identifier to indicate which user group applies. The letter “C” precedes the CUG identifier.
- Reverse charging identifier: If the remote physical unit will be charged for the call (reverse charging), include the number one (“1”) in the DIALNO. The letter “R” precedes the “1.” With this identifier, you can override what is specified on the PORT definition statement.

The DIALNO specification, including separator characters (“P,” “C,” or “R”) and the optional user facility characters, can be up to 32 characters in length. Table 39 shows examples of DIALNO specifications.

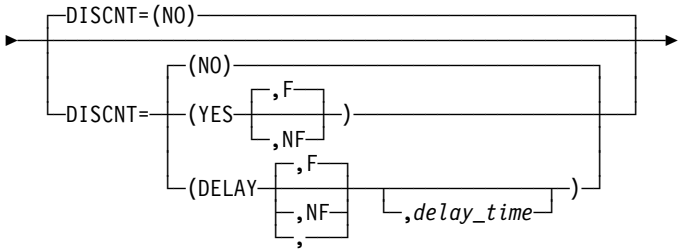
Table 39. Examples of DIALNO Specifications for the PATH Definition Statement

Example	Explanation
DIALNO=457031203	Address of a called physical unit.
DIALNO=457031203P0137	Address of a called physical unit with RPOA identifier of 0137.
DIALNO=457031203C31	Address of a called physical unit with CUG index of 31.
DIALNO=457031203R1	Address of a called physical unit with reverse charging indicated.
DIALNO=457031203C31R1	Address of a called physical unit with CUG index of 31 and reverse charging indicated.

Local Area Networks (VM, VSE)

DIALNO is not usually used in local area networks. However, for strictly call-out devices, DIALNO is required to enable VTAM to contact them.

DISCNT



statements: PU
dependency: none

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 physical unit.

DISCNT=(YES)
DISCNT=(NO)
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 broken (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 “Switched SDLC Connection” in the *VTAM 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.

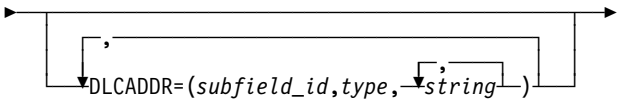
Each device has its own requirements regarding “final-use” status. To determine whether to code F or NF for a given device, consult the appropriate installation publication for the device.

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 of 1–255 is valid.

DISCNT=DELAY is valid for types 2 and 2.1 PUs.

Note: A delay time value can also be assigned for this PU through the MODIFY DEFAULTS operator command.

DLCADDR



statements: PATH
dependency: LINENM
range: See description

specifies the dial information used to initiate a connection with a physical unit over a switched link. DLCADDR can pass dial information in any of the following formats:

- Compressed alphanumeric
- Binary coded decimal
- EBCDIC
- Decimal
- Hexadecimal.

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 line 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 an indication of 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.

Note: DLCADDR is mutually exclusive with the LINENM operand for any single switched path definition statement.

Multiple DLCADDR entries 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)
```

The maximum number of PATH statements in a switched major node that can have DLCADDR coded is 32767.

Note: If you receive an error message referring to the first DLCADDR coded on a PATH statement, limited syntax checking is done on the subsequent DLCADDRs coded.

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 "Dependent LU Server" in the *VTAM 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.

The first DLCADDR entry defined for each PATH definition statement contains the information that identifies the DLC type, which can be one of the following:

- Data link switching
- Ethernet

- FDDI
- Frame relay
- Internal PU
- SDLC leased
- SDLC switched
- Token-ring
- X.25

The remaining DLCADDR entries define the DLC signaling information (addresses and dial digits) for each DLC type.

subfield_id
indicates the subfield. A value between 1–96 can be specified.

The first DLCADDR entry must have a *subfield_id* of 1. Also, each entry must have a unique *subfield_id*. If two DLCADDR entries coded have the same *subfield_id*, VTAM treats the two entries as duplicates. VTAM ignores the second entry and issues an error message.

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.

type
indicates the data format of the dial 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.

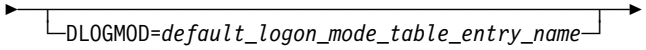
string
defines the dial information in the data format defined by *type*. Multiple *string* values can be coded for a *subfield_id type* pair. For example:

```
DLCADDR=(1,TYPE1,STRING1,STRING2,STRING3)
```

You can specify up to 250 bytes of dial information for a single DLCADDR entry. If coding multiple *string*s within a single entry, the total number of bytes available for dial information is reduced by 2 for each additional DLCADDR *string*.

For DLUR physical units, *string* defines either the DLC type or the DLC signaling information in the data format defined by *type*.

DLOGMOD

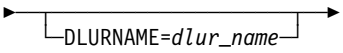


statements: PU, LU
dependency: PUTYPE=1 or 2

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 666.

DLURNAME



statements: PATH
dependency: 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.

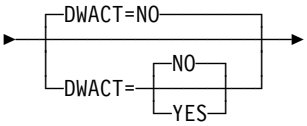
Note: If DLURNAME is specified, do not code the following operands:

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



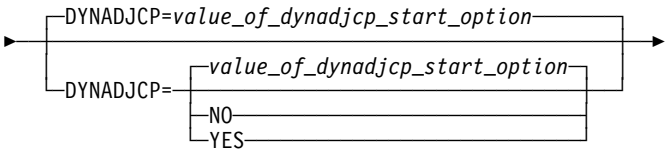
statements: PU
dependency: none

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.

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.

DYNADJCP



statements: PU
dependency: APPN only; ignored if CPNAME coded

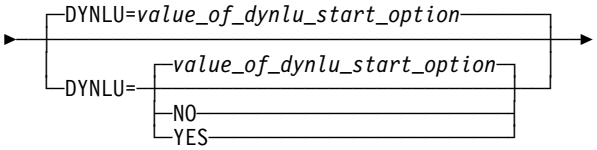
specifies whether an adjacent control point can be created dynamically for the adjacent node.

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



statements: PU
dependency: 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.

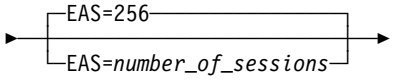
DYNLU=YES
specifies that dynamic allocation of CDRSC definitions is supported.

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 23.

EAS (VM, VSE)



statements: PU, LU
dependency: see note
range: 0–65535

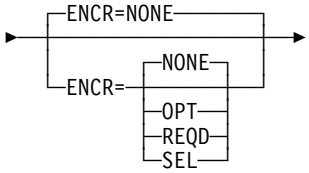
specifies the estimated number of sessions that will be active with this logical unit at any given time.

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 534 for information about BSBUF.

Note: EAS is valid only for an independent logical unit (LOCADDR=0). If you code it for any other logical units, VTAM ignores it. Also, EAS has meaning only when the PU connects through a channel-attached SDLC or packet-switched line.

ENCR (MVS)



statements: PU, LU
dependency: none

indicates whether this logical unit has any special requirements for enciphering and deciphering messages.

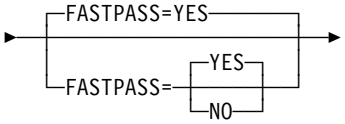
ENCR=NONE
specifies that this logical unit cannot engage in cryptographic sessions.

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.

FASTPASS (MVS)



statements: PU, LU
dependency: 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.

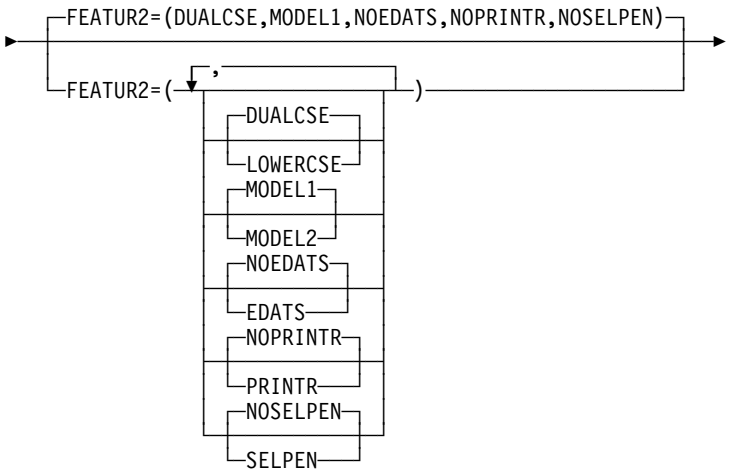
Some SLUs are incompatible with this enhancement. If your SLU is incompatible with this enhancement, code FASTPASS=NO.

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



statements: PU, LU
dependency: 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.

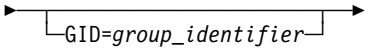
FEATUR2=MODEL1
FEATUR2=MODEL2
identifies the specific model number (Model 1 or 2) for this 3275, 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code MODEL2 for those devices that have a default screen or buffer size of 1920 bytes.

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 “INQUIRE-Obtain Logical Unit Information or Application Program Status” in *VTAM 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

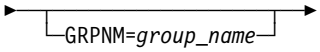


statements: PATH
dependency: none
range: 1–255

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



statements: PATH
dependency: CALL; VERID; VERIFY

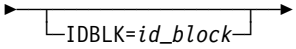
provides the name of a GROUP definition statement in the NCP or channel-attachment major node that defines a group of SDLC switched links. The line group must have all the characteristics necessary to process the telephone number and must be compatible with the type of physical unit.

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) or a 3172 Interconnect Controller group, *group_name* should be the name of the logical group definition.

IDBLK



statements: PU
dependency: 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).

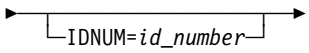
For any switched peripheral node (PUTYPE=1 or 2):

- If the node is not a type 2.1 peripheral node, you must code both IDBLK and IDNUM.
- If a type 2.1 peripheral node supports it, coding CPNAME is recommended, although you can code both IDBLK and IDNUM instead.

  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



statements: PU
dependency: 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.
- If a type 2.1 peripheral node supports it, coding CPNAME is recommended, although you can code both IDBLK and IDNUM instead.

  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.

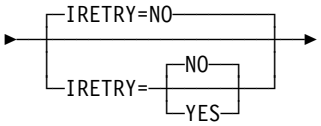
The node identifier field of an XID contains IDNUM along with a value of X'FFF' as the IDBLK.

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

IRETRY



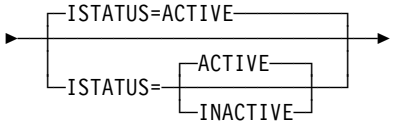
statements: PU
dependency: PUTYPE=1 or 2

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



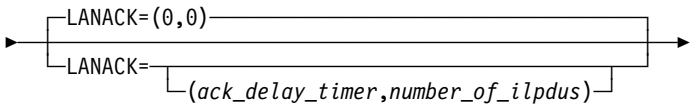
statements: PU, LU
dependency: none

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 "VARY ACT Command" in *VTAM Operation* for information on using the VARY ACT command.

LANACK (VM, VSE)



statements: PU
dependency: see note
range: 0–25.5 seconds (*timeout_value*)
0–127 (*number_of_ilpdus*)

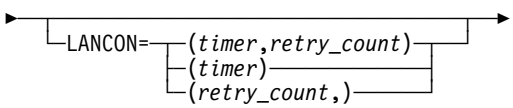
The operand specifies the acknowledgement delay timer (t2) and number (n3) of I-LPDUs received prior to sending an acknowledgement. If the timer expires, or if the count is reached, an acknowledgement is sent for any outstanding data.

The timer value is expressed in seconds or tenths of seconds. A value of 0 means that the timer is disabled.

You cannot specify 0 or take the default for just the timer or the counter. If you code one as 0, you must code 0 for both. For example, you cannot code LANACK=(0,2). If you want the timer to be 0, you must code LANACK=(0,0). Similarly, you cannot code LANACK=(3,0). If you want the counter to be 0, you must code LANACK=(0,0).

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or to a processor through its multi-protocol communication subsystem.

LANCON (VM, VSE)



statements: PU
dependency: see note
range: 0–25.4 seconds (*timeout_value*)
0–254 (*retry_count*)

specifies the timer (*ct1*) and the retry count (*cn2*) for the connection LPDUs.

ct1 specifies the time, in seconds or tenths of seconds, that elapses before a retry is attempted during connection or disconnection of a link station.

If you specify 0 or do not specify a value for the timer, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

cn2 specifies the number of times to retry a transmission during connection or disconnection of a link station.

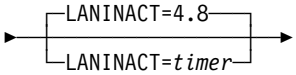
A value of 0 means that no retry will be attempted.

When the number of retries (*cn2*) has been reached and the timer (*ct1*) has elapsed for each retry, the link station is considered to be inoperative.

If you do not specify LANCON for this physical unit, the default values for the timer and the retry count are the values that you specify on the LANCON operand of the PORT definition statement in the LAN major node.

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANINACT (VM, VSE)



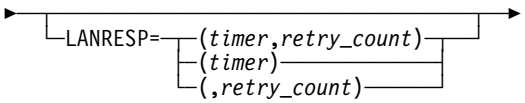
statements: PU
dependency: see note
range: 0–25.5 seconds (*timeout_value*)

specifies the timer inactivity (*t*) in seconds or tenths of seconds. *t* specifies the time used to determine an inactive condition on a link station.

If you specify 0, the timer is disabled.

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANRESP (VM, VSE)



statements: PU
dependency: see note
range: 0–25.4 seconds (*timeout_value*)
0–254 (*retry_count*)

specifies the timer (*t1*) and the retry count (*n2*) for a station in connected state.

t1 specifies the time, in seconds and tenths of seconds, that elapses before a retry is attempted when a station is connected.

If you specify 0, the IBM Token-Ring Subsystem Controller default timer value is used. Consult the appropriate documentation for your subsystem to determine the default timer value.

n2 specifies the number of times to retry a transmission while a station is connected.

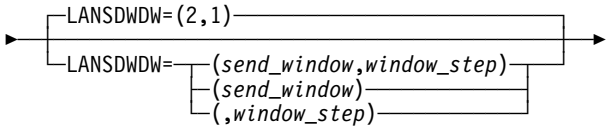
A value of 0 means that no retry will be attempted.

When the number of retries (*n2*) has been reached and the timer (*t1*) has elapsed for each retry, the link station is considered to be inoperative.

If you do not code the LANRESP operand for this physical unit, the default values for the timer and the retry count are the values that you specify on the LANCON operand of the PORT definition statement in the LAN major node.

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANSDWDW (VM, VSE)



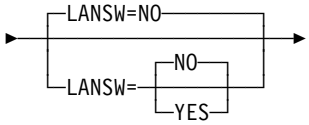
statements: PU
dependency: see note
range: 1–127 (*send_window*)
0–k (*send_window_step*)

specifies the send window (*k*) and window step (*nw*). The send window specifies the maximum number of sequentially numbered I-LPDUs that the link station can have outstanding at any time. The window step specifies the number of sequentially numbered I-LPDUs that a link station must receive before increasing the local window.

A value of 0 for the window step (*nw*) means that the send window (*k*) is static and dynamic windowing is disabled.

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LANSW (VM, VSE)



statements: PU
dependency: see note

specifies whether the physical unit can be used as a station on a LAN.

LANSW=NO

specifies that the physical unit cannot be used as a station on the LAN.

LANSW=YES

specifies that the physical unit can be used as a station on the LAN.

If you do not code this operand, you must code a value for one of the following operands for the physical unit to be used as a station on the LAN:

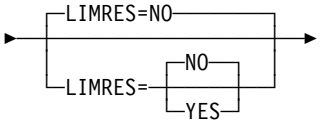
LANACK
LANCON
LANINACT
LANRESP
LANSDWDW
MACADDR
SAPADDR.

The minimum coding required for a dial-out physical unit is MACADDR=macaddr. The minimum coding required for a dial-in physical unit is LANSW=YES.

If a physical unit can be used on the LAN, additional storage is allocated to contain LAN definition values and their defaults when the physical unit is defined. This additional storage is 16 bytes for each physical unit that can be used on the LAN.

Note: This operand applies only if the physical unit can be attached directly to a processor through its IBM Token-Ring Subsystem Controller or a processor through its multi-protocol communication subsystem.

LIMRES



statements: PU
dependency: LIMQSINT; LU 6.2 only; PUTYPE=1 or 2

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.

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.

This operand applies only to LU 6.2 conversations and PU types 1 and 2.

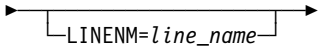
LIMRES=NO

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



statements: PATH
dependency: DIALNO; DLCADDR

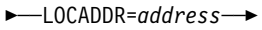
names a line for the direct call function.

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.

LOCADDR



statements: LU
dependency: none
range: 0–255 (PU Type 2)
0–63 (PU Type 1)

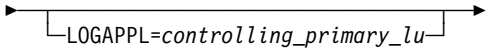
specifies the logical unit's local address at the physical unit. For dependent logical units, the range of valid local addresses depends on the PUTYPE of the associated physical unit. All dependent logical units under a given physical unit must have unique LOCADDR values.

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



statements: PU, LU
dependency: 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.

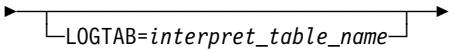
Notes:

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 (MVS) 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 “Automatic Logons” in the *VTAM Network Implementation Guide*.

LOGTAB

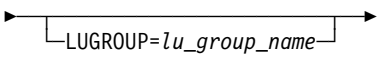


statements: PU, LU
dependency: PUTYPE=1 or 2

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 659 for a description of how to define an interpret table.

LOGTAB is valid for dependent logical units only.

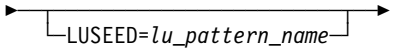
LUGROUP



statements: PU
dependency: 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



statements: PU
dependency: LUGROUP

provides a 1–8 character pattern name that the SDDLU exit routine uses to generate an LU name when dynamically defining an LU.

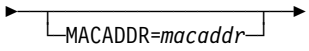
If you use the IBM-supplied SDDLU exit routine, this operand is required and must have the following format:

- 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 “Selection of Definitions for Dependent LUs (SDDLU) Exit Routine” in *VTAM Customization*.

This operand is valid only when you code the LUGROUP operand.

MACADDR (VM, VSE)



statements: PU
dependency: LAN

specifies the 12-digit hexadecimal medium access control (MAC) address for the station on the ring that this PU definition statement defines. If you specify the MAC address as an 8-digit hexadecimal number, VTAM prefixes the number with 4000 to form the 12-digit address.

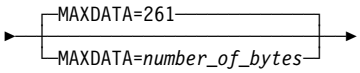
Note: The 12-digit value of MACADDR is the same as the last twelve digits of the DIALNO operand on the PATH definition statement in an NCP/Token-Ring interconnection (NTRI). See page 419 for an example of the format.

X'000000000000' is not a valid MACADDR valid.

This operand applies only if the physical unit can be attached directly to a processor with a local area network. MACADDR is required if the physical unit is used as a dial-out physical unit.

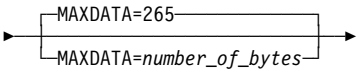
MAXDATA

For PU type 1:



statements: PU
dependency: see notes
range: 5-65535

For PU type 2:



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).

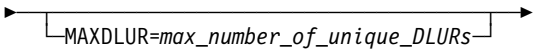
Note: The RU part of the PIU contains the user data. When VTAM segments the PIU, the middle and last PIUs will contain three more bytes of user data in the RU because there is no RH field. The value for the PIU varies with the device; for example, the 3276 requires MAXDATA=262.

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 or a 3172 controller. The type 2.1 node supplies the actual value used when the connection is established.

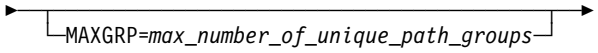
MAXDLUR



statements: VBUILD
dependency: 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.

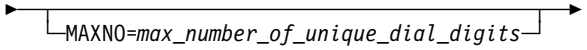
MAXGRP



statements: VBUILD
dependency: 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

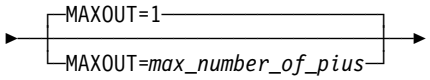


statements: VBUILD
dependency: PATH
range: 0–32766

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



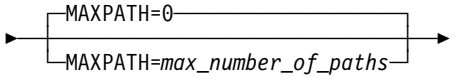
statements: PU
dependency: ignored for type 2.1 if attached through 3172
range: 1–127

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 a 3172 controller.

MAXPATH



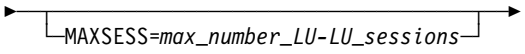
statements: PU
dependency: none
range: 0–256

specifies the number of dial paths defined for the physical unit. A value of 0 means that only dial-in paths are available.

Note: The number of path definition statements is limited to the value specified for MAXPATH.

Refer to the switched PATH definition statement on page 406 for a description of defining dial paths to a physical unit.

MAXSESS

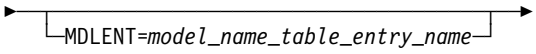


statements: PU, LU
dependency: independent LUs attached to an NCP
range: 1–65535

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

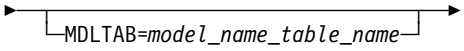


statements: PU, LU
dependency: MDLTAB; PUTYPE=1 or 2

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



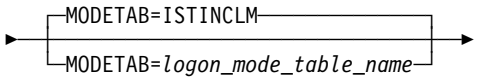
statements: PU, LU
dependency: PUTYPE=1 or 2

specifies the name of the model name table to be used for the logical unit. See “Model Name Table” on page 683 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 714 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 “MODIFY TABLE Command” in *VTAM Operation* for more information about the MODIFY TABLE command.

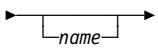
MODETAB



statements: PU, LU
dependency: 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 666. 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



statements: VBUILD, PATH

statements: 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.

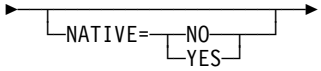
The optional *name* you give to this definition statement can be the same as the name VTAM gives to a major node and its definitions when they are filed in the VTAM definition library. The name VTAM gives to the major node is used when messages are issued regarding the node's status.

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. *name* must differ from *control_point_name* (CPNAME), or a sense code is given.

On LU: provides the required minor node name of the logical unit represented by this definition statement.

NATIVE



statements: PU
dependency: 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.

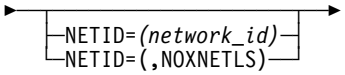
- If the NETIDs match, the connection defaults to a native connection.
- If the NETIDs are different, the connection defaults to a nonnative connection.

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



statements: PU
dependency: 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. MVS,VM It is required for a connection to another network. VSE NETID is not supported for cross-network connections.

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 Chapter 4, “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.

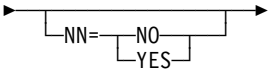
For dependent LU server PUs, VTAM will not dynamically process the network ID when CPSVRMGR sessions are established. To correctly define the NETID of a dependent LU server, if the NETID coded on the DLURNAME operand on the PATH statement is different from VTAM's network ID, the NETID parameter on the PU statement must be coded with the NETID coded on the DLURNAME otherwise the VTAM's network ID will be used for the PU.

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 should fail if the NETID sent in the bind from the LU is different than the NETID specified in VTAM, even though PU-PU connections can establish. If no NETID is included in the bind, or if the NETID 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



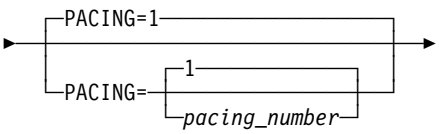
statements: PU
dependency: APPN only

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



statements: PU, LU
dependency: PUTYPE=1 or 2
range: 0–63

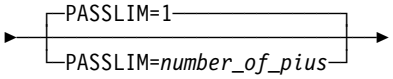
specifies how VTAM paces the flow of data from the boundary node, which performs pacing for a channel-attached SNA device, to the SLU. When the SLU and the PLU are in the same domain, the PACING value is ignored.

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 “Session Pacing” in the *VTAM Network Implementation Guide*.

PASSLIM

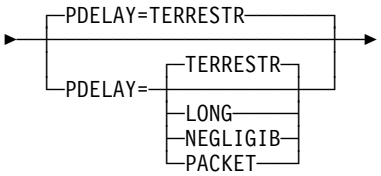


statements: PU
dependency: PUTYPE=1 or 2
range: 1–MAXOUT value

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



statements: PU
dependency: 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.

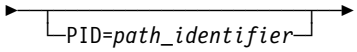
PDELAY=NEGLIGIB
indicates a local area network delay (less than .48 milliseconds).

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

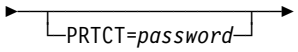


statements: PATH
dependency: none
range: 1–255

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



statements: PU
dependency: PUTYPE=4 or 5

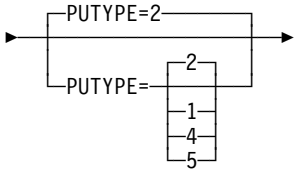
specifies a 1–8 EBCDIC character password used to verify the identity of a caller or receiver when switched subarea connections are being established. If you do not specify this operand, the identity of the caller or receiver is not verified when a connection is established.

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.

Note: VM, VSE PRTCT does not apply to X.21, X.25, or LAN connections.

PUTYPE



statements: PU
dependency: 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.

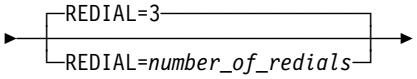
For switched subarea connections, code PUTYPE=4 or PUTYPE=5. PUTYPE=4 defines the physical unit as an NCP. PUTYPE=5 defines the physical unit as a VTAM.

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, 2, or 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 431 for a description of the station ID.

REDIAL



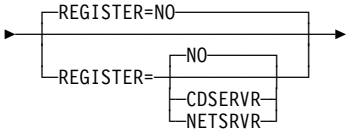
statements: PATH
dependency: none
range: 0–254

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. Specify 0 for X.25 communication.

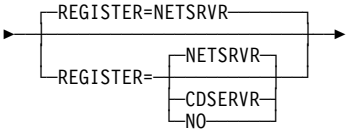
REGISTER

For Independent LUs:



statements: PU, LU
dependency: APPN only

For Dependent LUs:



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.

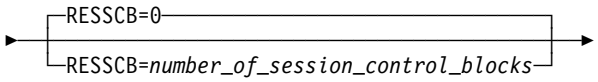
For dependent LUs, NETSRVR is the default, and LOCADDR must be specified as a non-zero value.

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.

RESSCB



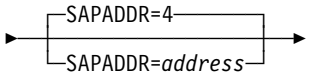
statements: LU
dependency: independent LUs
range: 0–65535

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 (VM, VSE)



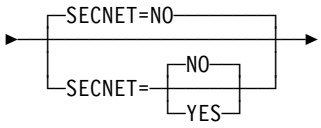
statements: PU
dependency: LAN
range: 4–252

specifies the service access point (SAP) address of the physical unit that is on the ring. The hexadecimal value of SAPADDR is equivalent to the third and fourth digits of the DIALNO operand of the PATH definition statement in an NCP/Token-Ring interconnection (NTRI) definition.

The value of *n* must be a multiple of 4.

This operand applies only if the physical unit can be attached directly to a processor with a local area network.

SECNET



statements: PU
dependency: PUTYPE=1 or 2

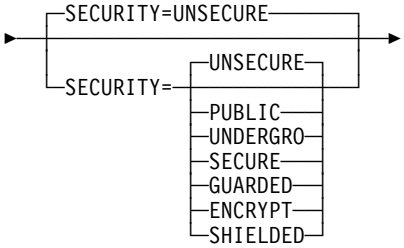
specifies whether this physical unit is associated with a secondary network containing resources whose connectivity is not defined to VTAM.

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 (MVS, VM)



statements: PU
dependency: 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.

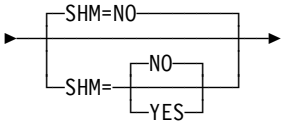
SECURITY=SECURE
specifies a secure conduit, not guarded.

SECURITY=GUARDED
specifies a guarded conduit, physical only.

SECURITY=ENCRYPT
specifies link encryption.

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded.

SHM



statements: PATH
dependency: DIALNO; GRPNM; LINENM; NPSI

specifies whether the switched subarea connection made using this PATH definition statement should be a short-hold mode connection.

VSE For boundary connections using the X.21 SHM/MPS feature, use the SHOLD operand to specify short-hold mode connections.

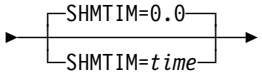
SHM=NO
specifies that short-hold mode is not to be used.

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.

This operand is mutually exclusive with the LINENM operand. It is valid only for connections established through X.25 Network Packet Switching Interface (NPSI).

SHMTIM



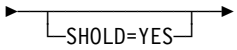
statements: PATH
dependency: NPSI connections
range: 0.0–6553.5 seconds

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.

SHOLD (VSE)

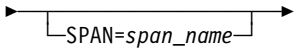


statements: PATH
dependency: DIALNO; GRPNM; LINENM

tells VTAM that this path is an X.21 short-hold mode/multiple port sharing (SHM/MPS) path. VTAM uses the values of *free* and *npoll* coded for the SHOLD operand on the corresponding GROUP definition statement for the channel-attachment major node, which is described on page 108.

If you code SHOLD=YES, also code DIALNO and GRPNM on this PATH definition statement but do not code LINENM.

SPAN

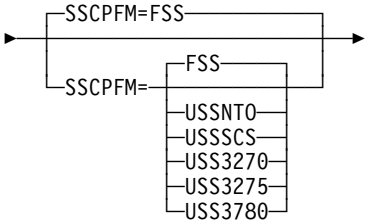


statements: PU, LU
dependency: NetView

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 *Installation and Administration Guide*.

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

SSCPFM



statements: PU, LU
dependency: 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.

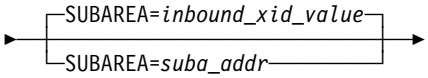
SSCPFM=USSSCS
specifies that character-coded RUs are supported.

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
is 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

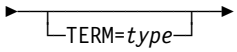


statements: PU
dependency: PUTYPE=4 or 5
range: 1–value of MXSUBNUM start option

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

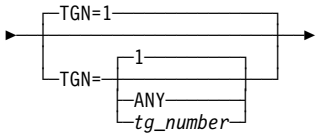


statements: PU, LU
dependency: PUTYPE=1 or 2

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 "Device Characteristics Field" in *VTAM Programming*.

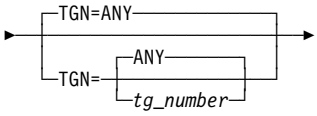
TGN

For subarea TGN:



statements: PU
dependency: PUTYPE=4,5
range: 1–255

For APPN TGN:



statements: PU
dependency: PUTYPE=2, CPNAME, NETID
range: 0–21

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.

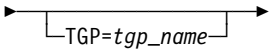
For APPN, coding TGN=0 means that the PU does not support parallel TGs.

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

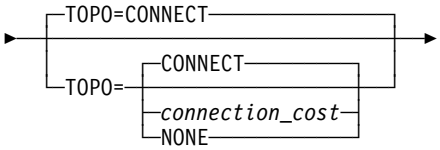


statements: PU
dependency: APPN only

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 495 for more information on transmission group profiles.

TOPO



statements: PU
dependency: 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.

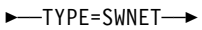
TOPO=*connection_cost*
indicates that the connection is to be treated as auto-active with a connection cost override. Auto-active allows a switched physical unit to be selected for routing prior to the actual establishment of the connection.

If a *connection_cost* is coded, then CPNAME, TGN, and NN 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.

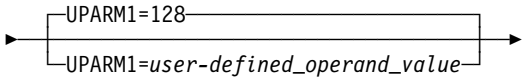
TYPE



statements: VBUILD
dependency: 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

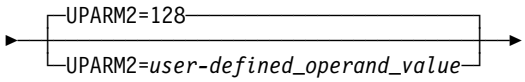


statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

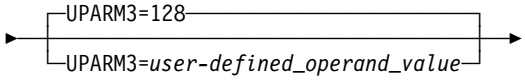
UPARM2



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

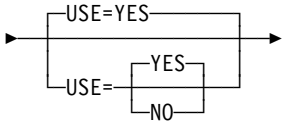
UPARM3



statements: PU
dependency: APPN only
range: 0–255

specifies a user-defined operand value.

USE



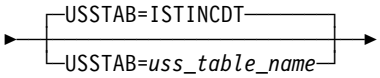
statements: PATH
dependency: none

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



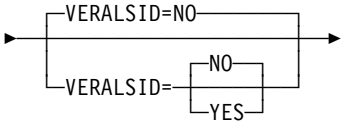
statements: LU
dependency: PUTYPE=1 or 2; dependent LUs only

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 690.

USSTAB is valid for dependent logical units only.

VERALSID



statements: PU
dependency: APPN only

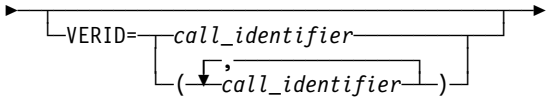
specifies whether the adjacent link station name should be used to determine the link station. When specified as YES, the connection endpoints must supply the same name in the CV0EF7 vector that is appended to the XID3. The name supplied in CV0EF7 is the name of a switched PU or the name of a PU under 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



statements: PATH
dependency: GRPNM

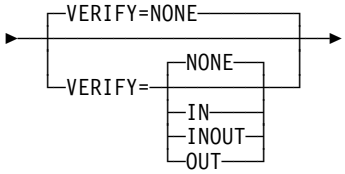
specifies one or more call identifiers to be used by call ID verification. Verification is provided only for X.25 subarea (type 4 and 5) connections.

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 in EBCDIC and can be up to 251 bytes long.

If more than one call identifier is specified, the list form of the VERID operand must be used. For example, to specify call identifiers *id1* and *id2*, code VERID=(id1,id2).

If you specify VERID, GRPNM is required. If you specify VERID, VERIFY=NONE is not valid.

VERIFY



statements: PATH
dependency: GRPNM; VERID

specifies when call ID verification is to be performed if this PATH definition statement is used. Verification is provided only for X.25 subarea (node types 4 and 5) connections. For X.25 switched connections, VERIFY=IN or VERIFY=NONE is valid.

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.

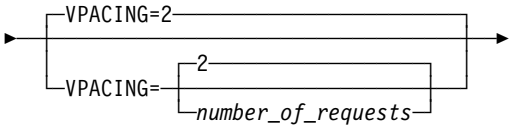
VERIFY=IN
specifies that call ID verification is to be performed for inbound calls using this PATH definition statement. If VERIFY=IN, CALL=OUT is not valid. If VERIFY=IN, VERID and GRPNM are required.

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



statements: PU, LU
dependency: 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 “Session Pacing” in the *VTAM Network Implementation Guide*.

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 connection that provides APPN host-to-host connectivity.

The transport resource list element is not a resource, but describes the connectivity characteristics of the multipath channel line that is being used by the APPN host-to-host connection. When an adjacent link station is activated, the TRLE operand on the PU definition statement identifies which TRLE definition statement VTAM uses to route data over the channel.

Refer to the *VTAM Network Implementation Guide* for more information about the APPN host-to-host channel function.

Table 40. Transport Resource List Major Node

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=TRL	462	R			
		CONFGDS	460				MVS,VSE
		CONFGPW	460				MVS,VSE CONFGDS
name	TRLE	LNCTL	460	R		MPC	
		READ	461	R			
		WRITE	462	R			
		MAXBFRU	460			1	
		REPLYTO	462			3.0	

Legend:

R Required operand.

Coding Notes

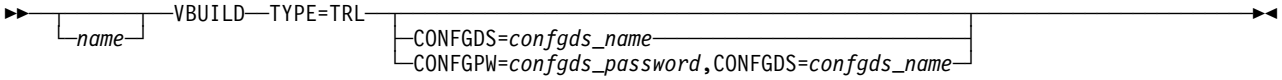
VBUILD Definition Statement: Code a VBUILD definition statement for each TRL major node.

TRLE Definition Statement: Code one TRLE definition statement for each multipath channel line that is used for APPN host-to-host connection in the TRL major node.

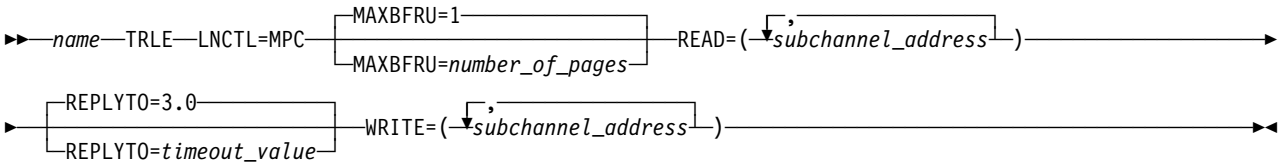
Full Syntax

The full syntax for the transport resource list major node. The syntax for each operand is repeated in the “Operand Description” section.

VBUILD



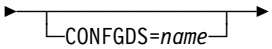
TRLE



Operand Descriptions

The following section describes the operands you can code for the transport resource major node.

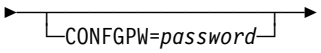
CONFGDS (MVS, VSE)



statements: VBUILD
dependency: 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 (MVS, VSE)

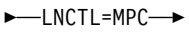


statements: VBUILD
dependency: CONFGDS

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.

LNCTL

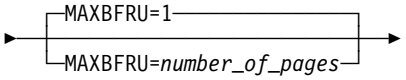


statements: TRLE

indicates that the link is a multipath channel-attachment link that can be used as an APPN host-to-host connection.

This operand is required.

MAXBFRU



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.

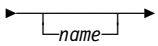
VTAM automatically substitutes a value of 16 for any coded value higher than 16 without issuing a warning message.

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.

name



statements: VBUILD

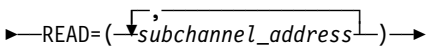
statements: TRLE

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.

READ



statements: TRLE
range: X'000'–X'FFFF'

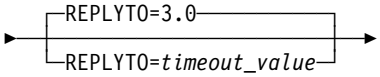
specifies the subchannel address (3-digit **VM,VSE** or 4-digit **MVS** hexadecimal number) used to read data from the adjacent host. You must code at least one READ subchannel. If you code an address that is not defined to the system, the activation fails for the entire group.

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, 128-30, 13A) defines the addresses 132, 128, 129, 12A, 12B, 12C, 12D, 12E, 12F, 130, and 13A. If any address in the range has not been defined to the system, the activation fails.

For each address on the READ operand, code the corresponding address on the WRITE operand in the adjacent host to provide a complete path. If you do not code the corresponding address, the entire group is deactivated.

Do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

REPLYTO

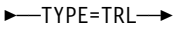


statements: TRLE
range: 0.1–25.5

specifies how long VTAM waits for completion of an multipath channel (MPC) 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 host is disabled for long intervals.

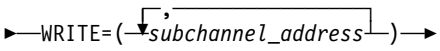
TYPE



statements: VBUILD

specifies that the VBUILD statement defines a TRL major node to VTAM.

WRITE



statements: TRLE
range: X'000'–X'FFFF'

specifies the subchannel address (3-digit **VM,VSE** or 4-digit **MVS** hexadecimal number) used to write data to the adjacent host. If you code an address that is not defined to the system, the activation fails for the entire group. Code at least 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. If any address in the range has not been defined to the system, the activation fails.

For each address on the WRITE operand, code the corresponding address on the READ operand in the adjacent host to provide a complete path. If you do not code the corresponding address, the entire group is deactivated.

Do not code the same subchannel address for the READ operand and the WRITE operand in the same host.

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

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
- Border node COS mapping definitions
- 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 41. Definition Statement for Routes

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	PATH	DESTSA	467	R			MXSUBNUM
		DELETER	467				see description
		ER0–ER15	467				
		VR0–VR7	468				
		VRPWS00–72	468				

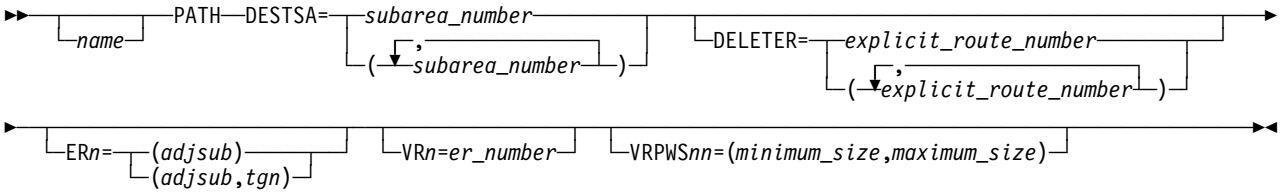
Legend:

R Required operand.

Full Syntax

The full syntax for VTAM routes follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

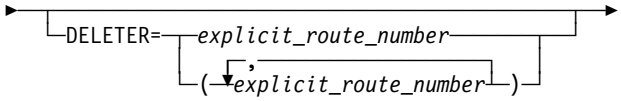
PATH



Operand Descriptions

The following section describes the operands you can code for VTAM routes.

DELETER



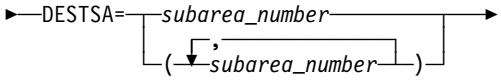
statement: PATH
dependency: see note

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

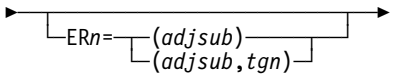


statement: PATH
dependency: MXSUBNUM

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



statement: PATH
dependency: 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.

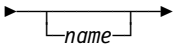
At least one ER must be specified on the PATH definition statement.

adjsub
is 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
is an optional transmission group number for the explicit route being defined.

If you do not specify a value, the default is 1.

name

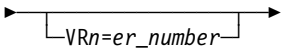


statement: PATH
dependency: none

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



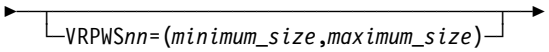
statement: PATH
dependency: none

associates the virtual route with the explicit route.

er_number is the explicit route number to which the virtual route is mapped.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72



statement: PATH
dependency: none

specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition.

Note that 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 “Virtual Route Window Sizes” in the *VTAM 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 “Dynamic Path Update” in the *VTAM 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 42. Definition Statements for Dynamic Update NCP Paths

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
ncpname	NCPPATH	NETID	473	R			
name	PATH	DESTSA	471	R			
		DELETER	471				see description
		ER0–ER15	472				
		VR0–VR7	473				
		VRPWS00–72	473				

Legend:

R Required operand.

Coding Notes

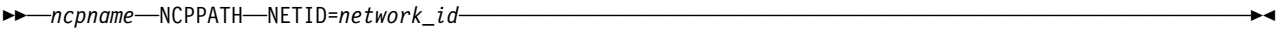
NCPPATH Definition Statement: Code the NCPPATH definition statement to identify the NCP and the network to which the path update specification set applies in a dynamic path update member. This definition statement is required, and must precede the path update specifications for each network supported by the NCP.

PATH Definition Statement: Code a PATH definition statement to define routes.

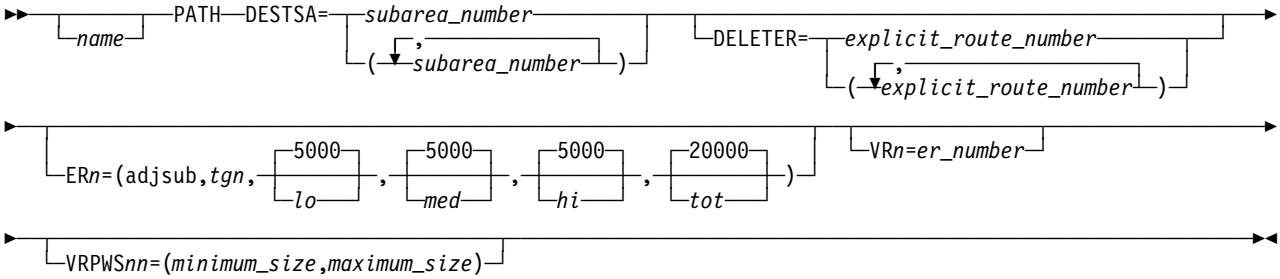
Full Syntax

The full syntax for dynamic update NCP paths follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

NCPPATH



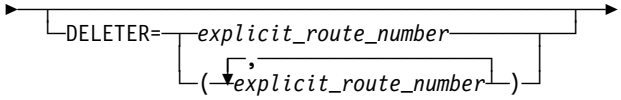
PATH



Operand Descriptions

The following section describes the operands you can code for dynamic update NCP paths.

DELETER



statement: PATH
dependency: see notes

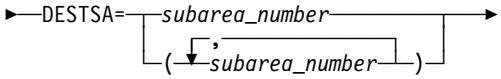
explicit_route_number is the explicit route number(s) for the route(s) to be deleted. The value of *explicit_route_number* can be ER0–ER15. Up to 16 explicit route numbers can be specified in DELETER.

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.

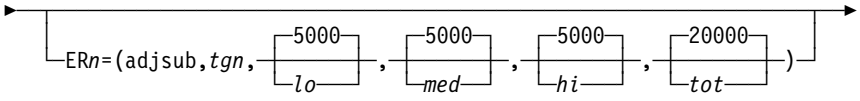
DESTSA



statement: PATH
dependency: 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



statement: PATH
dependency: 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.

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.

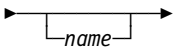
lo specifies the NCP's transmission group flow-control threshold for low priorities.

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

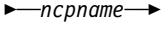


statement: PATH
dependency: none

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

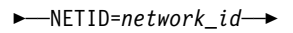


statement: NCPPATH
dependency: must match PU name of NCP

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



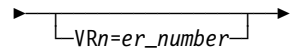
statement: NCPPATH
dependency: none

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



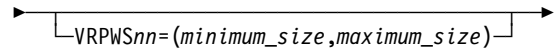
statement: PATH
dependency: none
range: 0–15

associates the virtual route with the explicit route.

er_number specifies the explicit route number to which the virtual route is mapped, and is a decimal integer in the range 0–15.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72



statement: PATH
dependency: none

specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition.

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 “Virtual Route Window Sizes” in the *VTAM Network Implementation Guide*.

Dynamic Path Update (VTAM)

To define a dynamic path specification set for VTAM, code VPATH and PATH definition statements.

Table 43. Definition Statements for Dynamic Update VTAM Paths

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
sscpname	VPATH	NETID	476	R			
name	PATH	DESTSA	475	R			
		DELETER	475				see description
		ER0–ER15	476				
		VR0–VR7	477				
		VRPWS00–72	477				

Legend:

R Required operand.

Coding Notes

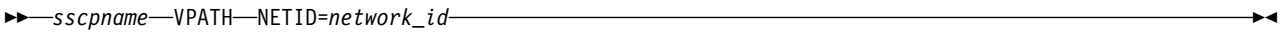
The operands on this PATH definition statement are the same as those on “VTAM Routes” on page 466, with one exception. If the SSCPNAME on the VPATH definition statement matches VTAM's, the values of the destination subarea and adjacent subareas used in the PATH definition statement must not exceed the 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.

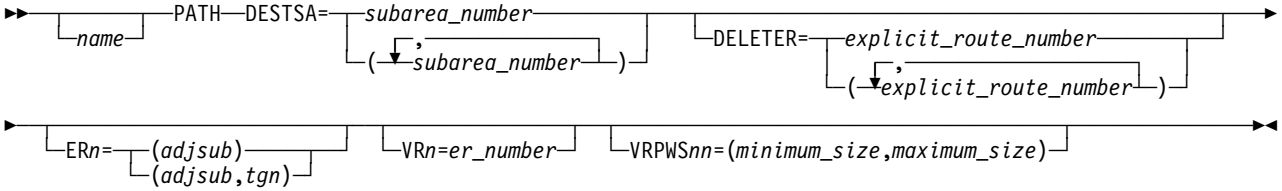
Full Syntax

The full syntax for dynamic update VTAM paths follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VPATH



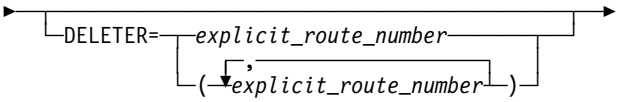
PATH



Operand Descriptions

The following section describes the operands you can code for dynamic update VTAM paths.

DELETER



statement: PATH
dependency: see notes

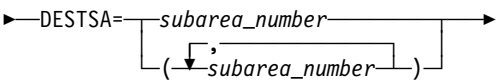
specifies the explicit route number(s) for the route(s) to be deleted. The value of *explicit_route_number* can be ER0–ER15. Up to 16 explicit route numbers can be specified in DELETER.

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.

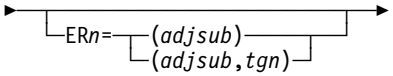
DESTSA



statement: PATH
dependency: 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



statement: PATH
dependency: none

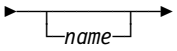
names the adjacent subarea and optionally specifies the transmission group (TG) number for the associated explicit routes originating in the host and leading to the destination subarea.

At least one ER must be specified on the PATH definition statement.

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

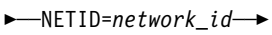


statement: PATH
dependency: none

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



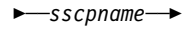
statement: VPATH
dependency: none

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.

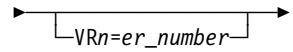
sscpname



statement: PATH
dependency: 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

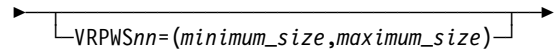


statement: PATH
dependency: none
range: 0–15

associates the virtual route with the explicit route. *er_number* specifies the explicit route number to which the virtual route is mapped, and is a decimal integer in the range 0–15.

At least one VR must be specified on the PATH definition statement.

VRPWS00-VRPWS72



statement: PATH
dependency: none

specifies the pacing window size for a certain virtual route and transmission priority. The virtual route must be defined within this path definition.

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 “Virtual Route Window Sizes” in the *VTAM 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 "Dynamic Reconfiguration and Change of Operands" in the *VTAM 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 10. 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 481 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 44. 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	484	R			
name	ADD	TO DRTYPE	483 482	R		 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 481 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.

Table 45. 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	484	R			
name	DELETE	FROM	483	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 481 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 46 (Page 1 of 2). 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	484	R			
name	MOVE	FROM	483	R			
		TO	483	R			
name	PU	ACTIVATE	482			NO	
		ADDR	482				

Table 46 (Page 2 of 2). Definition Statements for Dynamic Reconfiguration with VARY DRDS (MOVE Operation)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
Legend:							
R	Required operand.						

Coding Notes

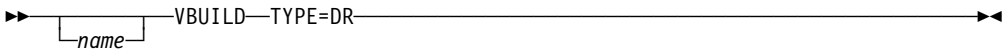
VBUILD Definition Statement: Code a VBUILD definition statement as the first statement in a group of dynamic reconfiguration definition statements.

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.

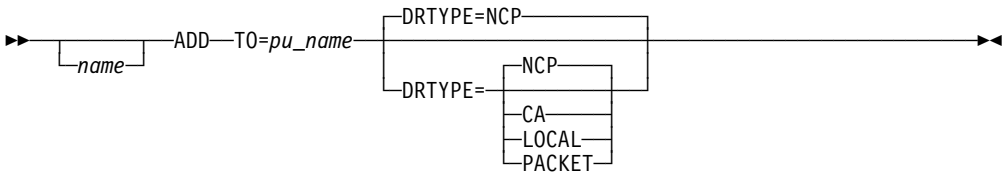
Full Syntax

The full syntax for dynamic reconfiguration with VARY DRDS follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

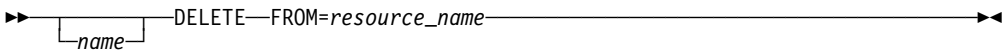
VBUILD



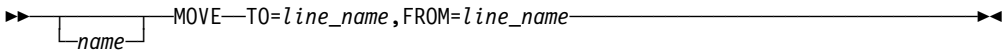
ADD



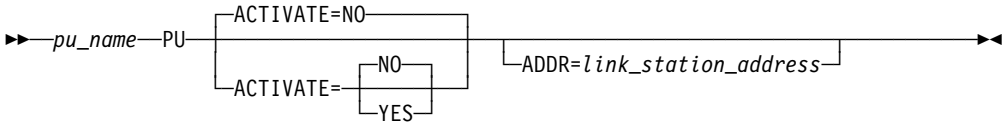
DELETE



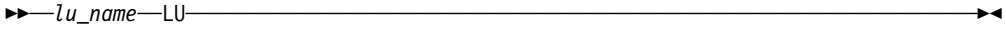
MOVE



PU



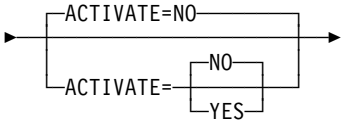
LU



Operand Descriptions

The following section describes the operands you can code for dynamic reconfiguration with VARY DRDS.

ACTIVATE

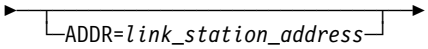


statements: PU
dependency: none

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

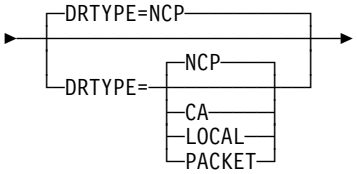


statements: PU
dependency: none
range: X'01'–X'FE'

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



statements: ADD
dependency: Adding LUs to PUs only

specifies the major node you are dynamically reconfiguring. Valid values are:

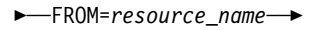
NCP an NCP major node

LOCAL a local SNA major node

CA an SDLC integrated communication adapter

PACKET a packet major node.

FROM



statements: DELETE, MOVE
dependency: none

resource_name can be any of the following:

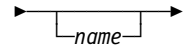
line name identifies a line from which you want to delete or move a physical unit. The physical units attached to this line must be types 1, 2, or 2.1. The line must be defined as nonswitched, and can be active or inactive. This operand is required if you are deleting or moving a physical unit. The line 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.

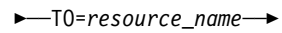
name



statements: VBUILD, ADD, DELETE, MOVE
dependency: 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



statements: ADD, MOVE
dependency: 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.

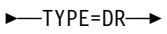
PU name
identifies a physical unit to which you want to add any associated logical units. The physical unit must be type 1, type 2, or type 2.1. This operand is required to identify the physical unit.

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



statements: VBUILD
dependency: 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 (MVS 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 47. Definition Statements for ADJSSCP Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=ADJSSCP	490	R			
name	NETWORK	NETID	489				
listname	ADJLIST		489				
sscpname	CDRM		490				
cdrmname	ADJCDRM		488				

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.

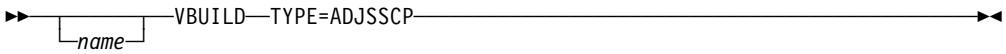
ADJCDRM Definition Statement: Code consecutive ADJCDRM definition statements to create a list of default SSCPs.

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.

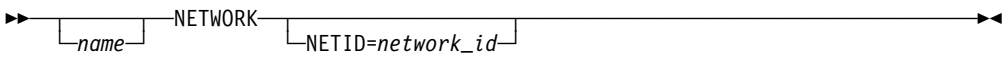
Full Syntax

The full syntax for the adjacent SSCP table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

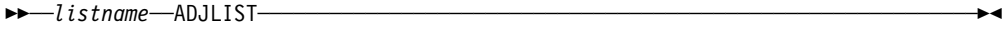
VBUILD



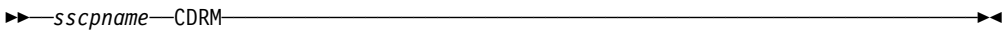
NETWORK



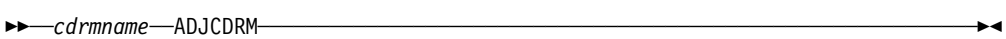
ADJLIST



CDRM



ADJCDRM



Operand Descriptions

The following section describes the operands you can code for the adjacent SSCP table.

cdrmname

statements: ADJCDRM
dependency: SORDER and SSCPORD start options

is 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 or SORDER=SUBAREA 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 SORDER=APPN or SORDER=SUBAREA, VTAM searches the adjacent SSCP table in the following order, with ISTAPNCP being added either after step 3b, if SORDER=APPN, or after step 3d, if SORDER=SUBAREA (ISTAPNCP is not added again if it is added after step 3b.):
   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. ISTAPNCP is not added unless it was included in one of the user-defined tables.

listname

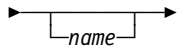
statements: ADJLIST
dependency: none

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.

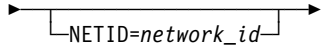
name



statements: VBUILD, NETWORK
dependency: 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

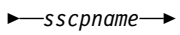


statements: NETWORK
dependency: none

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.

sscpname

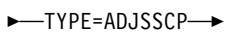


statements: CDRM
dependency: must match SSCPNAME start option

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



statements: VBUILD
dependency: 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 48. Definition Statements for ADJCLUST Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=ADJCLUST	494	R			
name	NETWORK	BNDYN	492			BNDYN start option value	BN=YES
		NETID	494				
name	NEXTCP	CPNAME	493				*this node's cpname*
		SNVC	494		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.

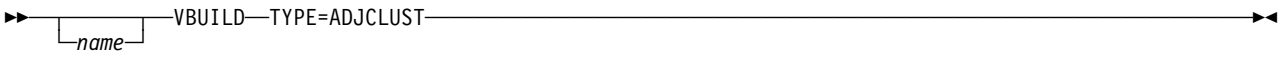
NETWORK Definition Statement: Code a NETWORK definition statement for each NETID or group of NETIDs that you want to define routing.

NEXTCP Definition Statement: Code a NEXTCP definition statement for each node that is to be searched.

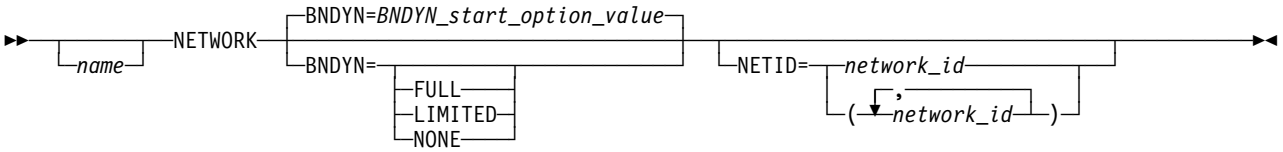
Full Syntax

The full syntax for the adjacent cluster routing definitions follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

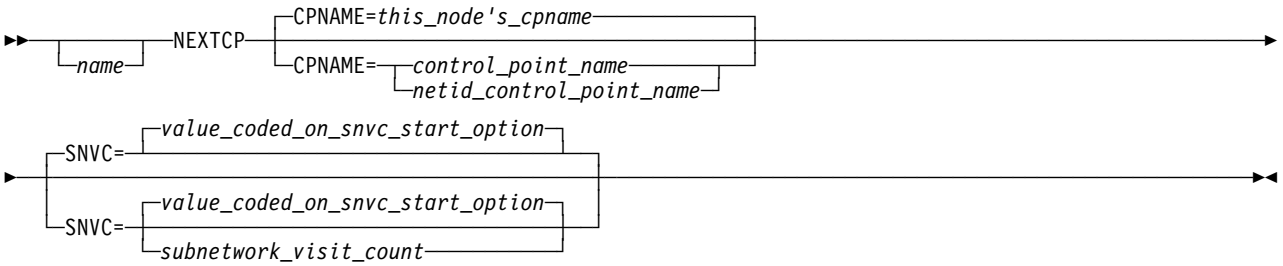
VBUILD



NETWORK



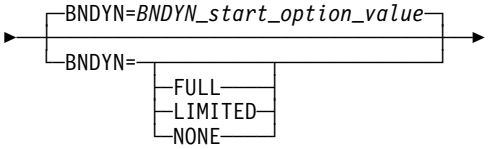
NEXTCP



Operand Descriptions

The following section describes the operands you can code for the adjacent cluster routing definition statements.

BNDYN



statements: NETWORK
dependency: BN=YES start option coded

specifies how VTAM will add nodes dynamically to the routing list.

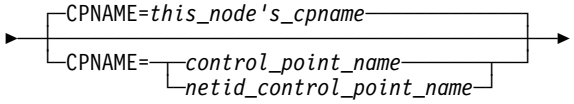
BNDYN=NONE
No nodes are added except those explicitly identified by NEXTCP definition statements.

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



statements: NEXTCP
dependency: none

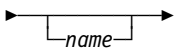
specifies the CP name of the node to which the search request is to be forwarded.

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



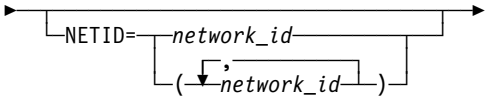
statements: VBUILD, NETWORK, NEXTCP
dependency: none

assigns a name to the definition statement.

To avoid confusion, use the same value for *name* and

- CPNAME when the NEXTCP definition statement is coded.
- NETID when the NETID definition statement is coded.

NETID



statements: NETWORK
dependency: none

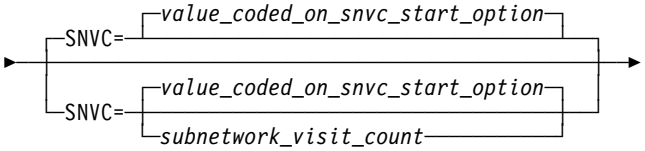
specifies the optional 1–8 character identifier of the destination network.

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



statements: NETWORK, NEXTCP
dependency: 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

statement: VBUILD
dependency: 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

MVS The IBM-supplied transmission group profiles are shipped with the name IBMTGPS in SYS1.ASAMPLIB. Copy IBMTGPS into SYS1.VTAMLST at VTAM installation.

VM The IBM-supplied transmission group profiles are shipped with the name IBMTGPS VTAMLST on the LOCALSAM DISK (2c2).

VSE The IBM-supplied transmission group profiles are shipped with the name IBMTGPS in the VSE/VTAM production sublibrary.

See “APPN Transmission Group (TG) Profile Definitions (IBMTGPS)” on page 811 for samples of the profiles, and refer to “Key Data Sets Containing Information for VTAM” in the *VTAM Network Implementation Guide* for more details.

Table 49. Definition Statement for Transmission Group Profile (TGP)

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	TGP	CAPACITY	496			8K	
		COSTBYTE	499			0	
		COSTTIME	499			0	
		PDELAY	499			TERRESTR	
		SECURITY	499			UNSECURE	MVS,VM
		UPARM1	500			128	
		UPARM2	500			128	
		UPARM3	500			128	

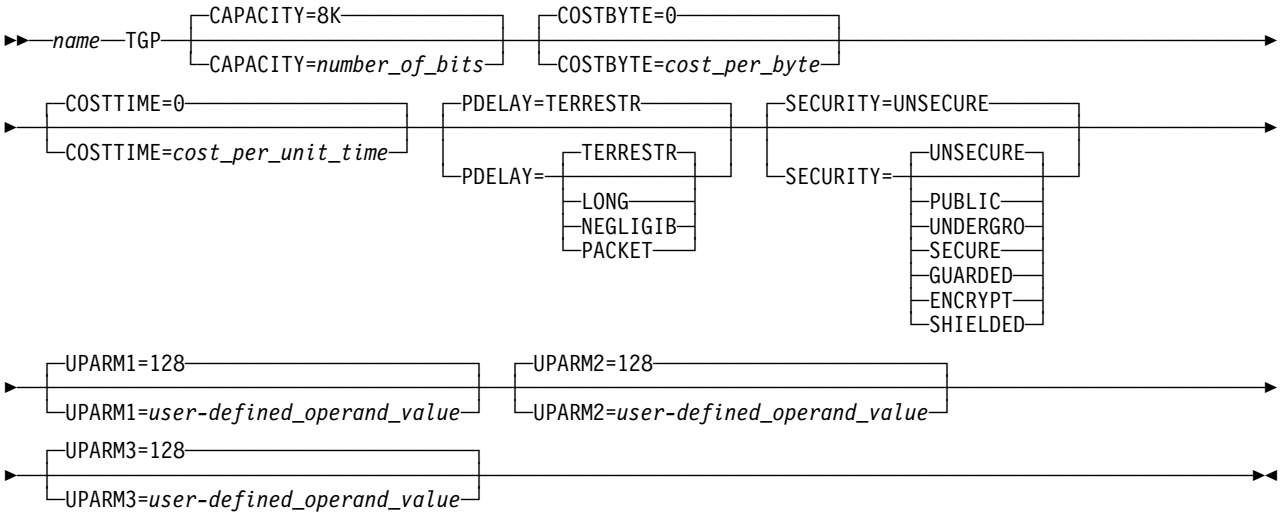
Legend:

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.

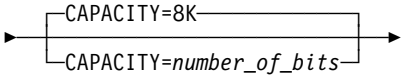
TGP



Operand Descriptions

The following section describes the operands you can code for the APPN transmission group profile.

CAPACITY



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 50 on page 497 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 50 (Page 1 of 2). 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
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

Table 50 (Page 1 of 2). CAPACITY Values When Displayed

VALUE CODED	D NET, TGP	D NET, TOPO
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
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

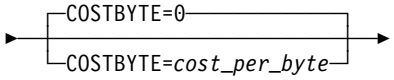
Table 50 (Page 2 of 2). CAPACITY Values When Displayed

VALUE CODED	D NET, TGP	D NET, TOPO
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
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

Table 50 (Page 2 of 2). CAPACITY Values When Displayed

VALUE CODED	D NET, TGP	D NET, TOPO
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

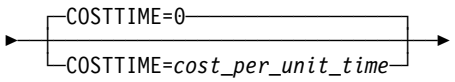
COSTBYTE



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.

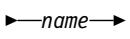
COSTTIME



range: 0–255

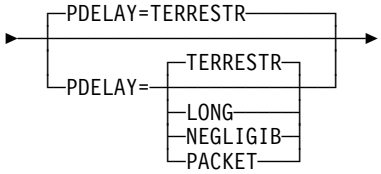
specifies a cost-per-unit-of-time to be associated with the TG, where 0 is the least expensive cost per unit of time and 255 is the most expensive.

name



specifies the required name of this TG profile.

PDELAY



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.

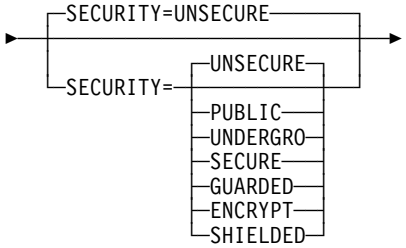
PDELAY=LONG
indicates a satellite delay (greater than 245.76 milliseconds).

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 (MVS, VM)



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

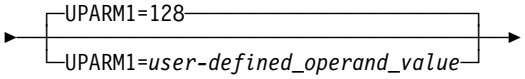
SECURITY=SECURE
specifies a secure conduit, not guarded

SECURITY=GUARDED
specifies a guarded conduit, physical only

SECURITY=ENCRYPT
specifies link encryption

SECURITY=SHIELDED
specifies a guarded conduit, physical and radiation shielded

UPARM1

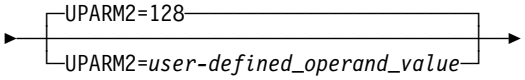


range: 0–255

specifies a user-defined operand value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

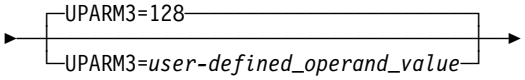
UPARM2



range: 0–255

specifies a user-defined operand value.

UPARM3



range: 0–255

specifies a user-defined operand value.

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 2 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 2. 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 51. Definition Statements for COSMAP Table

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=BNCOSMAP	503	R			
name	NETWORK	NETID	503				
cos_name	MAPSTO	COS	502				

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.

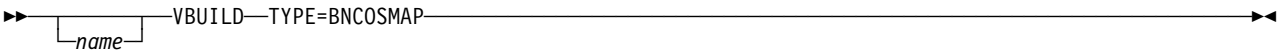
NETWORK Definition Statement: Code a NETWORK definition statement for each network for which you are defining a border node COS mapping table.

MAPSTO Definition Statement: Code a MAPSTO definition statement to define the native and nonnative APPN COS mappings.

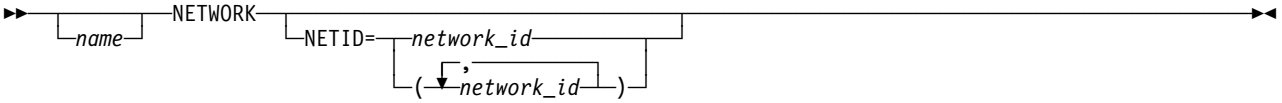
Full Syntax

The full syntax for the border node COS mapping definition follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

VBUILD



NETWORK



MAPSTO

Operand Descriptions

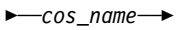
The following section describes the operands you can code for COS mapping tables.

COS

statements: MAPSTO
dependency: none

specifies the 1–8 character name identifying a native COS name.

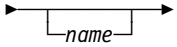
cos_name



statements: MAPSTO
dependency: none

specifies the 1–8 character name identifying the nonnative COS name.

name

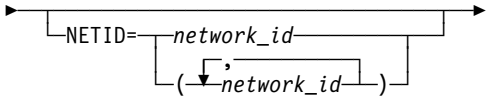


statements: VBUILD, NETWORK
dependency: none

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



statements: NETWORK
dependency: 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

statements: VBUILD
dependency: 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.

Table 52. Definition Statement for Network Node Server List

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	VBUILD	TYPE=NETSRVR	506	R			
		ORDER	505			FIRST	
name	NETSRVR	NETID	505				
		SLUINIT	506			REQ	

Legend:

R Required operand.

Coding Notes

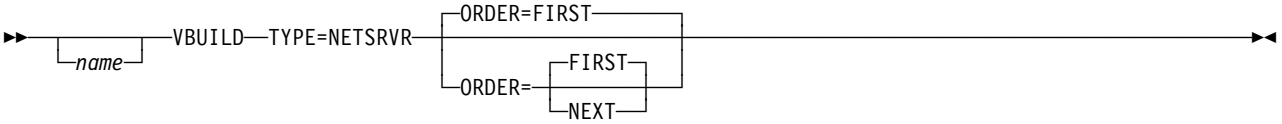
VBUILD Definition Statement: Code a VBUILD definition statement to begin the network node server list.

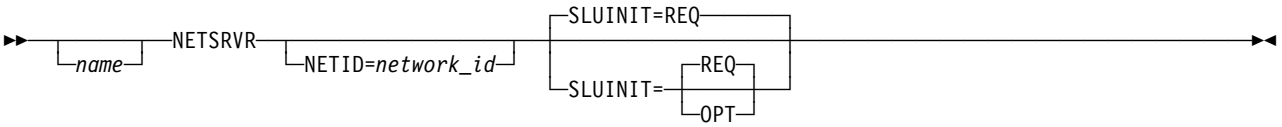
NETSRVR Definition Statement: Code a NETSRVR definition statement to define a network node (NN) as a network node server to the EN. There is an unlimited number of NETSRVR definition statements in any list (until VTAM runs out of storage). Duplicate entries in the list are ignored.

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.

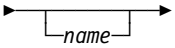




Operand Descriptions

The following section describes the operands you can code for the network node server list.

name

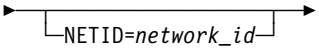


On VBUILD: assigns an optional name to a network node server list.

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.

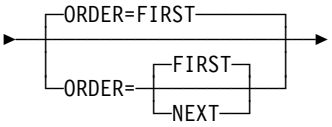
NETID



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

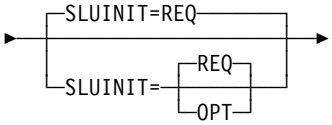


determines how the network node server list is processed.

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

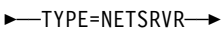


specifies whether the network node server is required to support SLU-initiated sessions as a prerequisite for establishment of CP-CP sessions between the indicated network node and this end node.

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



means that this VBUILD definition statement defines a network node server list.

Chapter 4. Start Options

About This Chapter

This chapter describes the VTAM start options and how to code them.

VM Start options for VSCS are coded as operands on the DTIGEN macroinstruction. The DTIGEN macroinstruction and its operands are described in Chapter 5, "VM/SNA Console Support (VSCS)" on page 603.

Refer to "Start Options" in the *VTAM 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 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,                                                              X
SSCPNAME=SSCP1A,                                                        X
NETID=NETA,                                                             X
DYNASSCP=YES,                                                           X
SSCPDYN=YES,                                                            X
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:

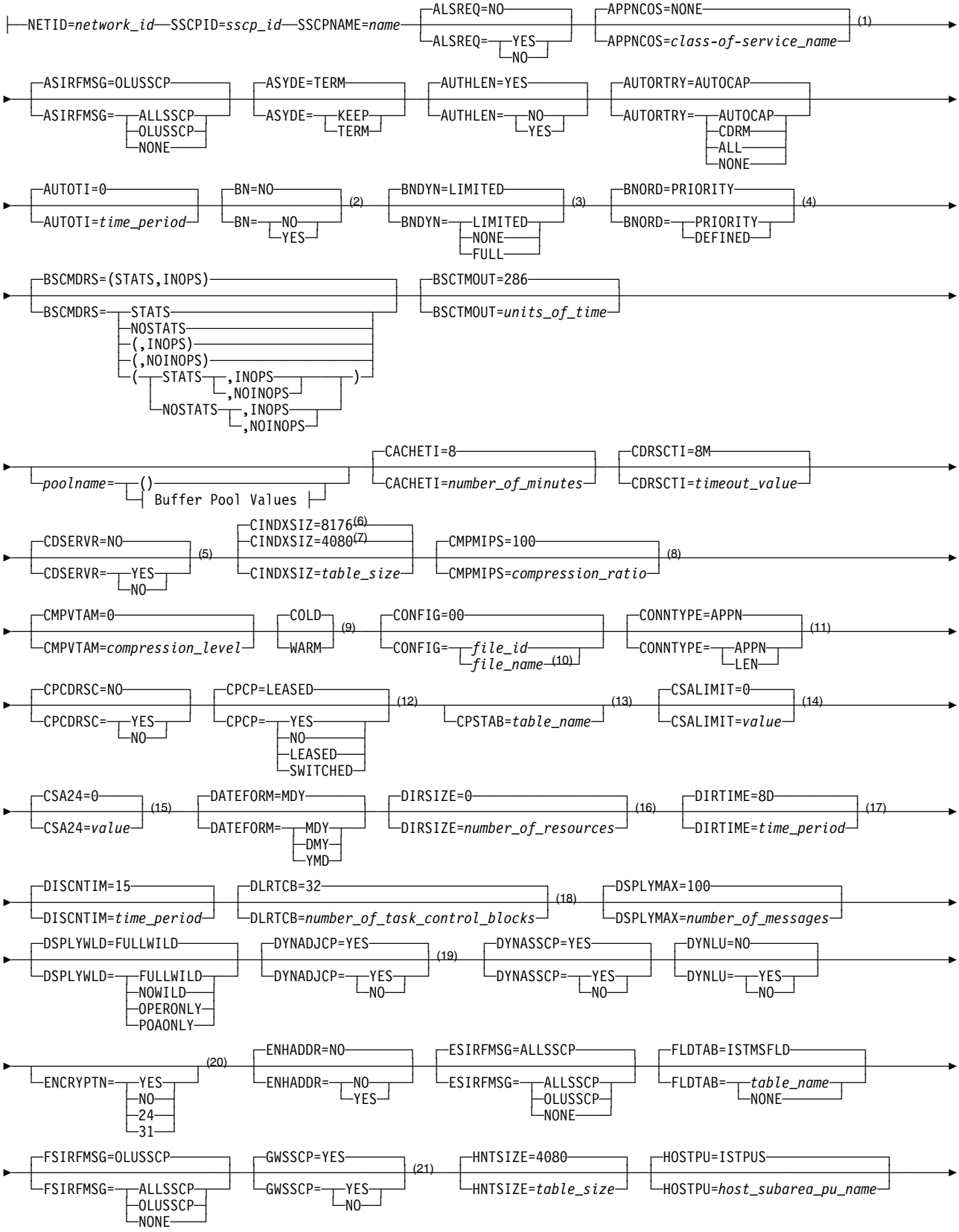
- Every record, except comment records and the last record, must have a nonblank character in position 72.
- You can code data through position 71, even if you are in the middle of an item. The remainder of the item can be continued in the next record. The

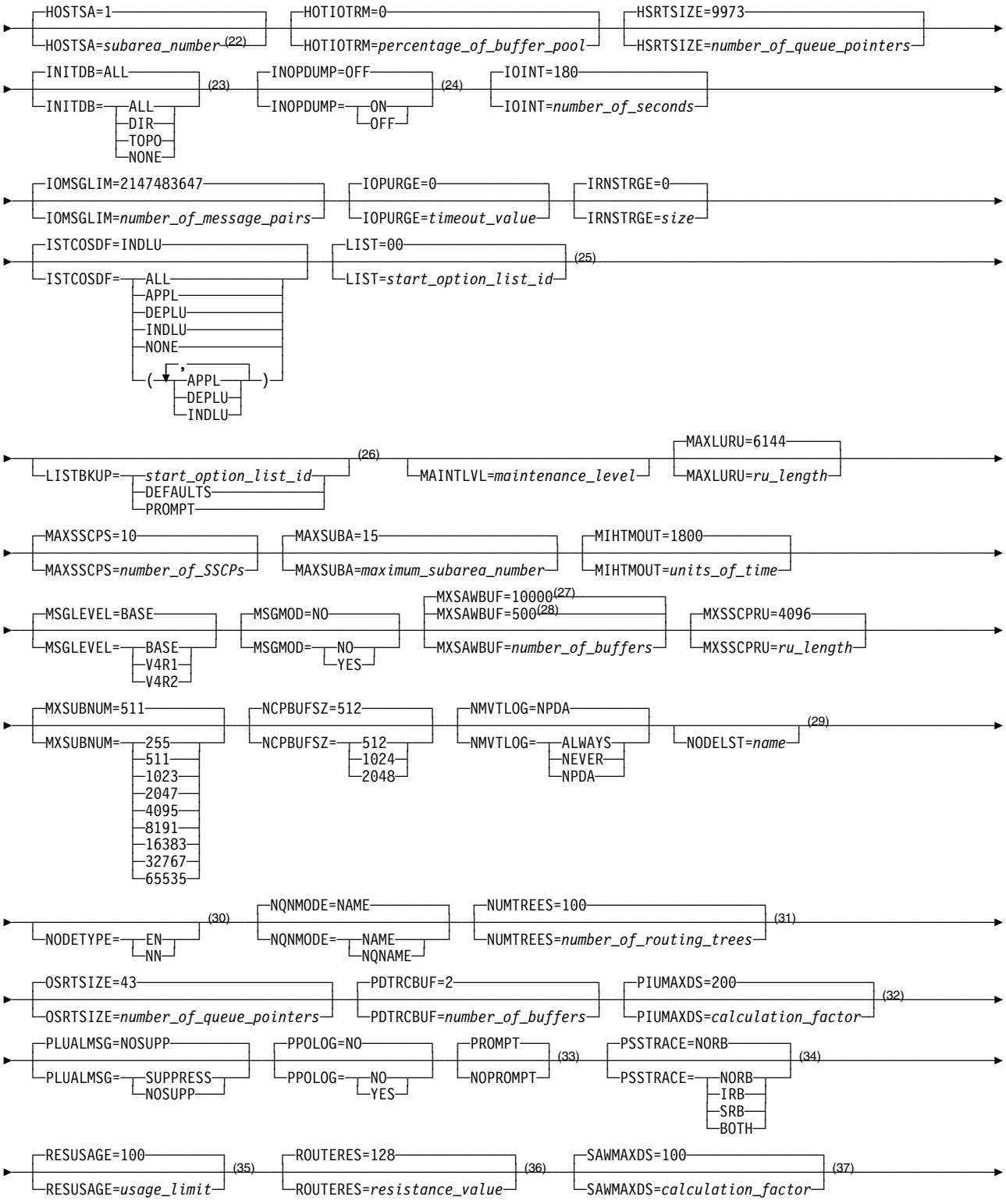
TRACE start option is an exception to this rule. TRACE and its qualifiers must be coded on one line.

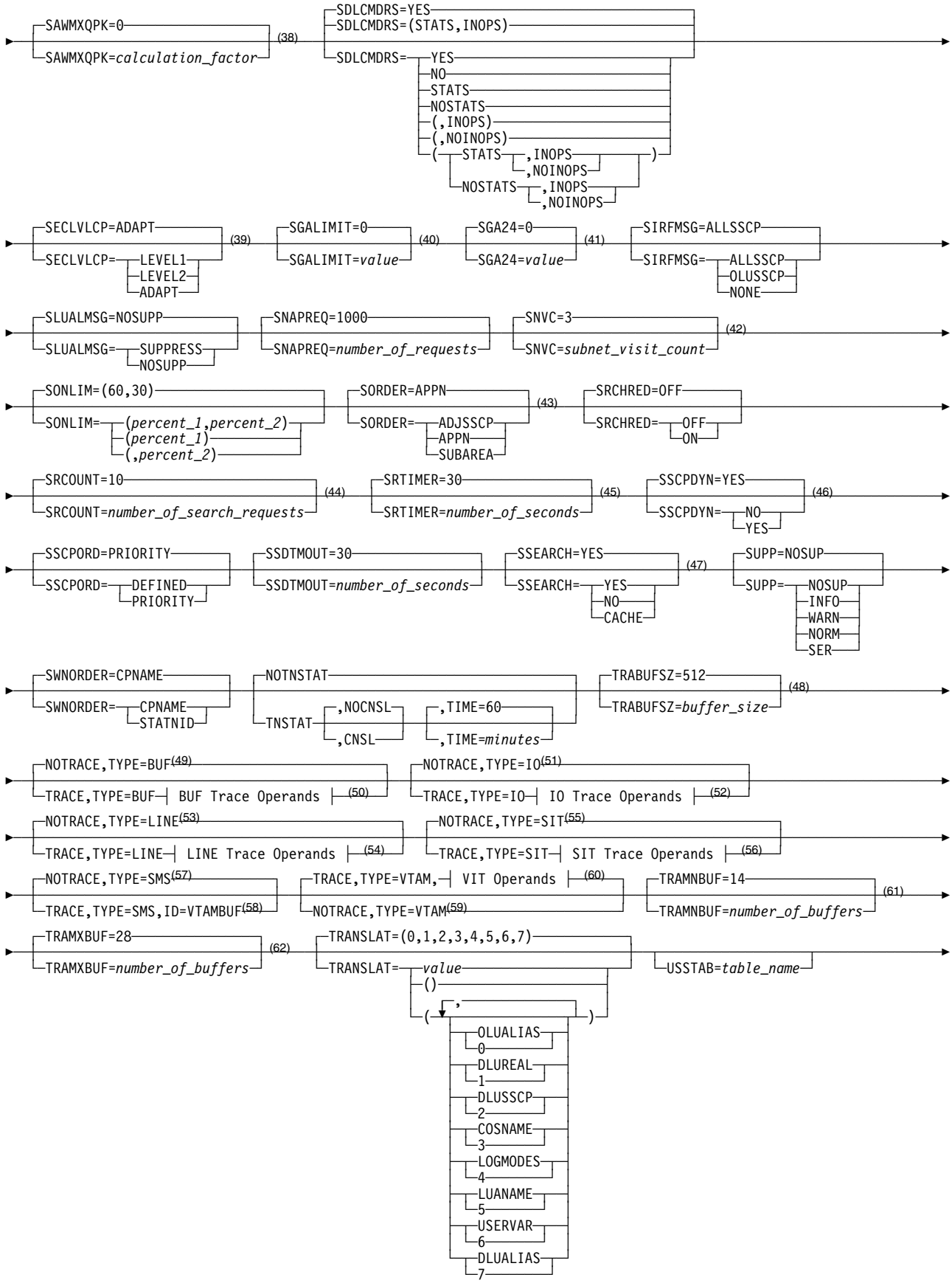
- If you do not code data through position 71, end the data in one record with an item, followed by a comma, and begin the next record with a new item.
- To create a comment line, enter an asterisk (*) in position 1 of any record other than a continuation record. A comment line cannot follow a continuation line.

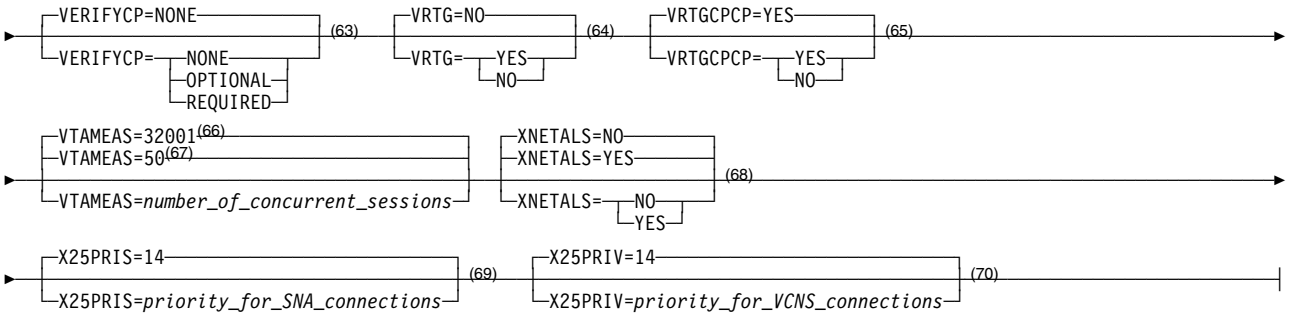
Start Options Syntax Diagrams

Options:

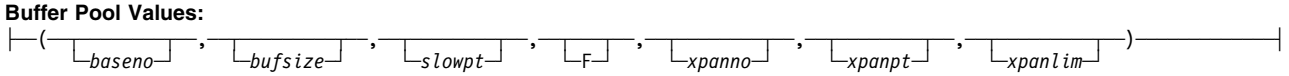



HOSTSA=1
HOSTSA=subarea_number (22)
HOTIOTRM=0
HOTIOTRM=percentage_of_buffer_pool
HSRTSIZE=9973
HSRTSIZE=number_of_queue_pointers
INITDB=ALL
INITDB= ALL DIR TOPO NONE (23)
INOPDUMP=OFF
INOPDUMP= ON OFF (24)
IOINT=180
IOINT=number_of_seconds
IOMSGLIM=2147483647
IOMSGLIM=number_of_message_pairs
IOPURGE=0
IOPURGE=timeout_value
IRNSTRGE=0
IRNSTRGE=size
ISTCOSDF=INDLU
ISTCOSDF= ALL APPL DEPLU INDLU NONE (, APPL DEPLU INDLU)
LIST=00
LIST=start_option_list_id (25)
LISTBKUP= start_option_list_id DEFAULTS PROMPT (26)
MAINTLVL=maintenance_level
MAXLURU=6144
MAXLURU=ru_length
MAXSSCPS=10
MAXSSCPS=number_of_SSCPs
MAXSUBA=15
MAXSUBA=maximum_subarea_number
MIHTMOUT=1800
MIHTMOUT=units_of_time
MSGLEVEL=BASE
MSGLEVEL= BASE V4R1 V4R2
MSGMOD=NO
MSGMOD= NO YES
MXSAWBUF=10000 (27)
MXSAWBUF=500 (28)
MXSAWBUF=number_of_buffers
MXSSCPRU=4096
MXSSCPRU=ru_length
MXSUBNUM=511
MXSUBNUM= 255 511 1023 2047 4095 8191 16383 32767 65535
NCPBUFSZ=512
NCPBUFSZ= 512 1024 2048
NMVTLOG=NPDA
NMVTLOG= ALWAYS NEVER NPDA
NODELST=name (29)
NODETYPE= EN NN (30)
NQNMODE=NAME
NQNMODE= NAME NQNAME
NUMTREES=100
NUMTREES=number_of_routing_trees (31)
OSRTSIZE=43
OSRTSIZE=number_of_queue_pointers
PDTRCBUF=2
PDTRCBUF=number_of_buffers
PIUMAXDS=200
PIUMAXDS=calculation_factor (32)
PLUALMSG=NOSUPP
PLUALMSG= SUPPRESS NOSUPP
PPOLOG=NO
PPOLOG= NO YES
PROMPT
NOPROMPT (33)
PSSTRACE=NORB
PSSTRACE= NORB IRB SRB BOTH (34)
RESUSAGE=100
RESUSAGE=usage_limit (35)
ROUTERES=128
ROUTERES=resistance_value (36)
SAWMAXDS=100
SAWMAXDS=calculation_factor (37)

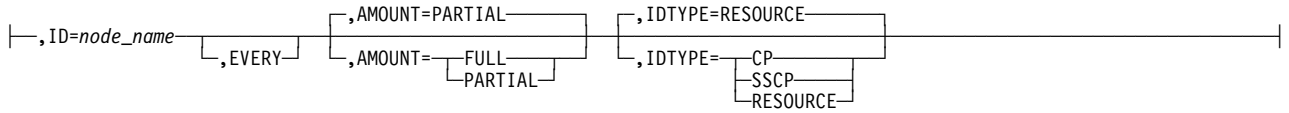

SAWMXQPK=0
SAWMXQPK=calculation_factor
(38)
SDLCMDRS=YES
SDLCMDRS=(STATS,INOPS)
SDLCMDRS=
YES
NO
STATS
NOSTATS
(,INOPS)
(,NOINOPS)
(STATS ,INOPS ,NOINOPS)
NOSTATS ,INOPS ,NOINOPS
SECLVLCP=ADAPT
SECLVLCP=
LEVEL1
LEVEL2
ADAPT
(39)
SGALIMIT=0
SGALIMIT=value
(40)
SGA24=0
SGA24=value
(41)
SIRFMSG=ALLSSCP
SIRFMSG=
ALLSSCP
OLUSSCP
NONE
SLUALMSG=NOSUPP
SLUALMSG=
SUPPRESS
NOSUPP
SNAPREQ=1000
SNAPREQ=number_of_requests
SNVC=3
SNVC=subnet_visit_count
(42)
SONLIM=(60,30)
SONLIM=
(percent_1,percent_2)
(percent_1)
(,percent_2)
SORDER=APPN
SORDER=
ADJSSCP
APPN
SUBAREA
(43)
SRCHRED=OFF
SRCHRED=
OFF
ON
SRCOUNT=10
SRCOUNT=number_of_search_requests
(44)
SRTIMER=30
SRTIMER=number_of_seconds
(45)
SSCPDYN=YES
SSCPDYN=
NO
YES
(46)
SSCPORD=PRIORITY
SSCPORD=
DEFINED
PRIORITY
SSDTMOUT=30
SSDTMOUT=number_of_seconds
SSEARCH=YES
SSEARCH=
YES
NO
CACHE
(47)
SUPP=NOSUP
SUPP=
NOSUP
INFO
WARN
NORM
SER
SWNORDER=CPNAME
SWNORDER=
CPNAME
STATNID
NOTNSTAT
TNSTAT
,NOCNSL
,CNSL
,TIME=60
,TIME=minutes
TRABUFSZ=512
TRABUFSZ=buffer_size
(48)
NOTRACE,TYPE=BUF
(49)
TRACE,TYPE=BUF
BUF Trace Operands
(50)
NOTRACE,TYPE=IO
(51)
TRACE,TYPE=IO
IO Trace Operands
(52)
NOTRACE,TYPE=LINE
(53)
TRACE,TYPE=LINE
LINE Trace Operands
(54)
NOTRACE,TYPE=SIT
(55)
TRACE,TYPE=SIT
SIT Trace Operands
(56)
NOTRACE,TYPE=SMS
(57)
TRACE,TYPE=SMS,ID=VTAMBUF
(58)
TRACE,TYPE=VTAM,
VIT Operands
(60)
NOTRACE,TYPE=VTAM
(59)
TRAMNBUF=14
TRAMNBUF=number_of_buffers
(61)
TRAMXBUF=28
TRAMXBUF=number_of_buffers
(62)
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
USSTAB=table_name



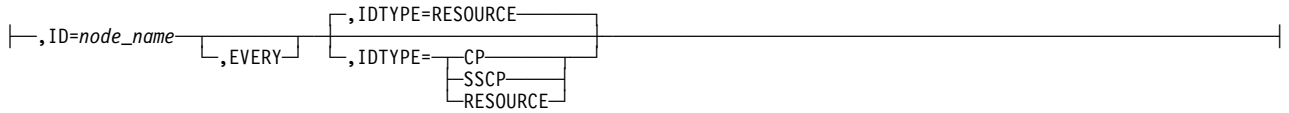
Buffer Pool Values:



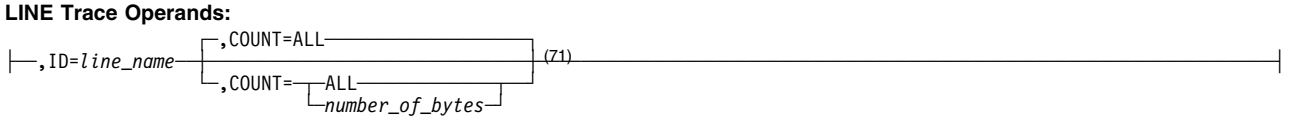
BUF Trace Operands:



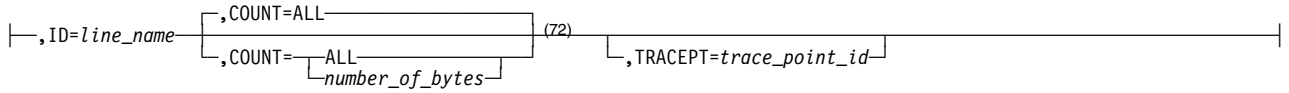
IO Trace Operands:



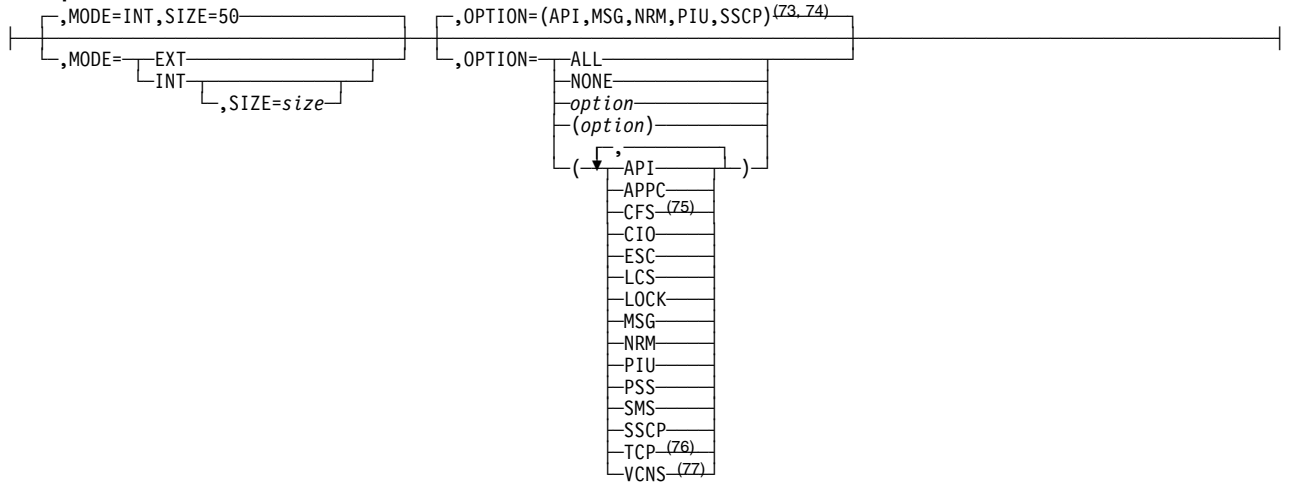
LINE Trace Operands:



SIT Trace Operands:



VIT Operands:



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 CINDXSIZ=8176 is the default for MVS and VM.

7 CINDXSIZ=4080 is the default for VSE.

[8] The CMPMIPS start option applies to MVS and VSE only, and is meaningful only if the value for CMPVTAM is greater than 1.

[9] The COLD and WARM start options apply to MVS and VSE only.

[10] CONFIG= *file_name* applies to MVS and VSE only.

[11] CONNTYPE is meaningful only if the NODETYPE start option is also used.

[12] CPCP is meaningful only if the NODETYPE start option is also used.

[13] The CPSTAB start option applies to VSE only.

[14] The CSALIMIT start option applies to MVS and VM only.

[15] The CSA24 start option applies to MVS and VM only.

[16] DIRSIZE is meaningful only if the NODETYPE=NN start option is also used.

[17] DIRTIME is meaningful only if the NODETYPE=NN start option is also used.

[18] The DLRTCB start option applies to MVS and VM only.

[19] DYNADJCP is meaningful only if the NODETYPE start option is also used.

[20] The ENCRYPTN start option applies to MVS only.

[21] The GWSSCP start option applies to MVS and VM only.

[22] HOSTSA provides subarea function. If you do not specify HOSTSA, but you specify NODETYPE, there is no default for HOSTSA and subarea function is not provided. If you do not specify HOSTSA and NODETYPE, a default of HOSTSA=1 is assumed and APPN function is not provided.

[23] INITDB is meaningful only if the NODETYPE=NN start option is also used.

[24] The INOPDUMP start option applies to MVS and VM only.

[25] 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.

[26] 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.

[27] MXSAWBUF=10000 is the default for MVS and VM.

[28] MXSAWBUF=500 is the default for VSE.

[29] The NODELST start option applies to MVS and VSE only.

[30] NODETYPE provides APPN function. The combination of NODETYPE and HOSTSA determines the configuration (subarea node, interchange node, migration data host, network node, or end node).

[31] NUMTREES is meaningful only if the NODETYPE=NN start option is also used.

[32] The PIUMAXDS start option applies to MVS only.

[33] 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.

[34] The PSSTRACE start option applies to MVS only.

[35] RESUSAGE is meaningful only if the NODETYPE=NN start option is also used.

[36] ROUTERES is meaningful only if the NODETYPE=NN start option is also used.

[37] The SAWMAXDS start option applies to MVS only.

[38] The SAWMXQPK start option applies to MVS only.

[39] The SECLVLCP start option applies to MVS and VM only, and is meaningful only if the NODETYPE and VERIFYCP start options are also used.

[40] The SGALIMIT operand applies to VSE only.

[41] The SGA24 operand applies to VSE only.

[42] SNVC is meaningful only if the BN=YES start option is also used.

[43] SORDER is meaningful only in a network node, interchange node, or migration data host.

[44] SRCOUNT is meaningful only if the SRCHRED=ON start option is also used.

[45] SRTIMER is meaningful only if the SRCHRED=ON start option is also used.

[46] The SSCPDYN start option applies only for interconnected networks (that is, GWSSCP=YES is used).

[47] SSEARCH is meaningful only if the NODETYPE=NN start option is also used.

[48] The TRABUFSZ start option applies to VM and VSE only.

[49] Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

[50] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[51] Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

[52] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[53] Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

[54] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[55] Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

[56] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[57] Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

[58] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[59] NOTRACE,TYPE=VTAM is accepted but ignored. Tracing is started with the default trace table size and the default options.

[60] Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

[61] The TRAMNBUF start option applies to VM and VSE only.

[62] The TRAMXBUF start option applies to VM and VSE only.

[63] The VERIFYCP start option applies to MVS and VM only, and is meaningful only if the NODETYPE start option is also used.

[64] VRTG is meaningful only if the NODETYPE and HOSTSA start options are also used.

[65] VRTGCPCP is meaningful only if the NODETYPE and HOSTSA start options are also used.

[66] VTAMEAS=32001 is the default for MVS and VM.

[67] VTAMEAS=50 is the default for VSE.

[68] The default for XNETALS is NO if BN=NO. The default for XNETALS is YES if BN=YES.

[69] The X25PRIS start option applies to VM only.

[70] The X25PRIV start option applies to VM only.

[71] COUNT applies only to the IBM 3720 and 3745 Communication Controllers.

[72] COUNT applies only to the IBM 3720 and 3745 Communication Controllers.

[73] The default options apply only to MODE=INT for MVS and VM.

[74] The default options apply to MODE=INT and MODE=EXT for VSE.

[75] The CFS trace option applies to MVS only.

[76] The TCP trace option applies to MVS only.

[77] The VCNS trace option applies to MVS and VM only.

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 53 on page 520)
- VTAM initialization (Table 54 on page 520)
- Storage (Table 55 on page 521)
- Session control (Table 56 on page 521)
- Connectivity (Table 57 on page 522)
- Session security (Table 58 on page 523)
- Messages (Table 59 on page 523)
- Recording and statistics (Table 60 on page 524)
- Traces and dumps (Table 61 on page 524)
- Buffer pools (Table 62 on page 524)
- Performance and tuning (Table 63 on page 524).
- APPN characteristics (Table 64 on page 526).

See “Descriptions of Start Options” on page 528 for more detailed descriptions of the start options.

SSCP or CP Characteristics Start Options

Table 53. Start Options That Define SSCP or CP Characteristics

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	MVS,VM Whether VTAM is a gateway SSCP
HOSTPU	The name of the host subarea PU
HOSTSA	The host subarea number
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
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
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 54 (Page 1 of 2). Start Options That Affect VTAM Initialization

Start Option	Description
COLD	MVS,VSE Upon 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
NODELST	MVS,VSE The name of the configuration restart file
NOPROMPT	Do not prompt the operator to enter start options
PROMPT	Prompt the operator to enter start options

Table 54 (Page 2 of 2). Start Options That Affect VTAM Initialization

Start Option	Description
WARM	MVS,VSE Upon starting VTAM, restore major nodes to their status prior to VTAM's last termination, as recorded in the configuration restart file

Storage Usage Start Options

Table 55. Start Options That Define Storage Usage

Start Option	Description
CSALIMIT	MVS,VM The maximum amount of common service area (CSA) that can be used by VTAM
CSA24	MVS,VM The maximum amount of 24-bit addressable CSA that can be used by VTAM
DLRTCB	MVS,VM 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
SGALIMIT	VSE The maximum amount of system GETVIS area (SGA) that can be used by VTAM
SGA24	VSE The maximum amount of 24-bit addressable SGA that can be used by VTAM

Session Control Start Options

Table 56 (Page 1 of 2). Start Options That Affect Session Control

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 performs cross-subnet searches
CDRDYN	Specifies whether the host is authorized to dynamically define CDRSC representations of resources
CMPMIPS	MVS,VSE 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
DYNADJCP	Whether dynamic entries are created in ISTADJCP for adjacent APPN nodes

Table 56 (Page 2 of 2). Start Options That Affect Session Control

Start Option	Description
DYNASSCP	Whether VTAM dynamically routes session establishment requests to all active adjacent SSCPs if no appropriate adjacent SSCP table is defined
DYNLU	Whether independent LUs can be dynamically defined
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
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 57 (Page 1 of 2). Start Options That Affect Connectivity. These start options affect session establishment, termination, or search order.

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
CPSTAB	**VSE** The name of the call progress signal (CPS) table used to control retry values
DISCNTIM	Amount of time VTAM delays deactivation of SSCP-PU session when there are no outstanding LU-LU session requests
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
VRTG	Whether virtual-route-based transmission groups connections are requested when SSCP-SSCP session is established

Table 57 (Page 2 of 2). Start Options That Affect Connectivity. These start options affect session establishment, termination, or search order.

Start Option	Description
VRTGCPCP	Whether CP-CP sessions are supported over the virtual-route-based transmission group
XNETALS	Whether VTAM can establish sessions with LUs in adjacent networks

Session Security Start Options

Table 58. Start Options That Affect Session Security

Start Option	Description
ENCRYPTN	MVS VTAM's level of cryptography support
SECLVLCP	MVS,VM Whether VTAM uses basic or enhanced protocol for session-level security verification during activation of LU 6.2 sessions involving APPN control points
VERIFYCP	MVS,VM Whether VTAM performs LU-LU session-level verification during activation of LU 6.2 sessions involving APPN control points

Messages Start Options

Table 59. Start Options That Affect Messages

Start Option	Description
ASIRFMSG	The SSCPs in which to issue messages IST890I and IST896I when an autologon session initiation request fails
DSPLYMAX	The maximum number of resources about which certain DISPLAY commands will provide information
DSPLYWLD	Whether wildcards are permitted in certain DISPLAY commands
ESIRFMSG	The SSCPs in which to 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
FSIRFMSG	The SSCPs in which to issue messages IST894I and IST895I 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 messages for session setup failures that occur because the PLU is unavailable
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 because the SLU is not enabled
SUPP	The message suppression level for the operator console

Recording and Statistics Start Options

Table 60. Start Options That Affect Recording and Statistics

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 61. Start Options That Affect Traces and Dumps

Start Option	Description
INOPDUMP	MVS,VM Whether to dump VTAM for certain INOP conditions
PSSTRACE	MVS 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 62. Start Options That Define Buffer Pools

Start Option	Description
APBUF	MVS The buffer pool that provides fixed common storage with 24-bit addressable buffers
BSBUF	The buffer pool that provides boundary session control block buffers
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
XDBUF	The buffer pool that provides XID buffers

Performance and Tuning Start Options

Table 63 (Page 1 of 3). Start Options That Affect Performance and Tuning

Start Option	Description
AUTOTI	Specifies how often pending automatic logon requests owned by this host are retried
BSCTMOUT	The time interval in which two general poll failures are allowed for BSC 3270 terminals

Table 63 (Page 2 of 3). Start Options That Affect Performance and Tuning

Start Option	Description
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
HSRTSIZE	The size of the SRT directory for the network containing the VTAM host node
IOINT	The time interval after which the VTAM operator is notified of outstanding I/O requests
IOPURGE	The time 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 time interval in which you can adjust the missing interrupt handling time 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
NUMTREES	The maximum number of APPN routing trees that can be saved
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	MVS A value used to calculate the maximum buffer limit for PIU data space
SAWMAXDS	MVS A value used to calculate the maximum buffer limit for SAW data space
SAWMXQPK	MVS 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
TRABUFSZ	VM,VSE The buffer size used for token-ring connections
TRAMNBUF	VM,VSE The minimum number of transmit buffers for token-ring connections

Table 63 (Page 3 of 3). Start Options That Affect Performance and Tuning

Start Option	Description
TRAMXBUF	VM,VSE The maximum number of transmit buffers for token-ring connections
VTAMEAS	The size of the table that contains information about sessions with an application program
X25PRIS	VM The connection priority for SNA-X.25 connections
X25PRIV	VM The connection priority for VCNS (VTAM Common Network Services)-X.25 connections

APPN Start Options

Table 64 (Page 1 of 2). Start Options That Define APPN Characteristics

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
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
INITDB	Whether to load the APPN directory database, topology database, neither, or both when VTAM is started
NODETYPE	The APPN node type (network node or end node)
NUMTREES	The maximum number of APPN routing trees that can be saved
ROUTERES	The relative amount of resistance to having this APPN node perform intermediate session routing
SECLVLCP	MVS,VM 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
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.

Table 64 (Page 2 of 2). Start Options That Define APPN Characteristics

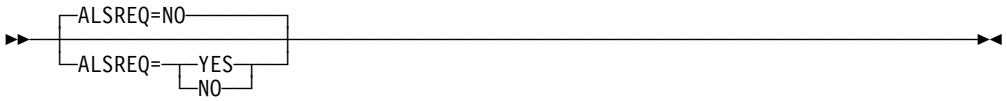
Start Option	Description
VERIFYCP	Whether VTAM performs LU-LU session-level verification during activation of LU 6.2 sessions involving APPN control points
VRTG	Whether virtual-route-based transmission groups connections are requested when SSCP-SSCP session is established
VRTGCPCP	Whether CP-CP sessions are supported over the virtual-route-based transmission group

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



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.

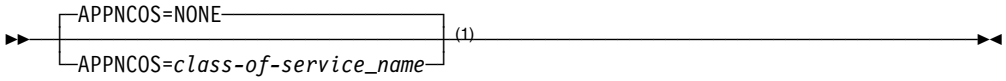
If adjacent link station names are required, only session requests over a matching adjacent link station in the list are allowed. For a LEN adjacent link station, this means the names must match. For an APPN adjacent link station, it means that another APPN adjacent link station (real or ISTAPNPU) is 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.

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



Note:
1 APPNCOS is meaningful only if the NODETYPE start option is also used.

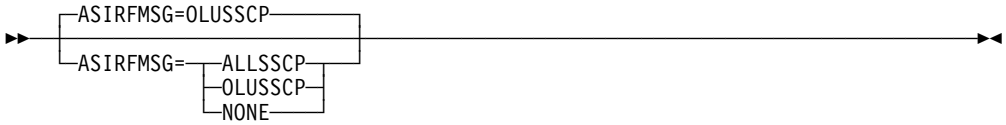
specifies the APPN class of service to be used if a requested class of service cannot be found in the topology and routing services class-of-service database. See “IBM-Supplied APPN Classes of Service” on page 654 for the names of the seven classes of service and “APPN Class-of-Service Definitions” on page 644 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



controls the display of messages IST890I and IST896I. These messages are issued when an autologon session initiation request fails.

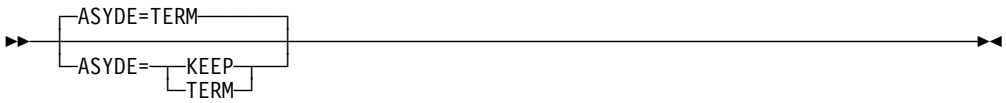
You can change the value of ASIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

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



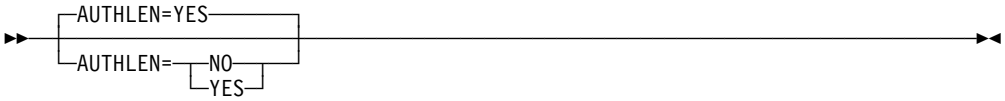
specifies whether to keep or terminate the session between the device and its application when VTAM receives an asynchronous device end from a local non-SNA device. For example, an asynchronous device end is caused when the device is powered off, or, for a printer, when the cover is lifted.

This option applies to all locally attached non-SNA devices except printers. See also “FEATUR2” on page 206 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

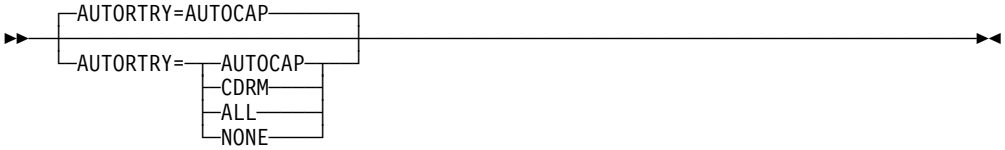


indicates whether VTAM will pass the transmission priority specified from one LEN node to another LEN node.

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



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.

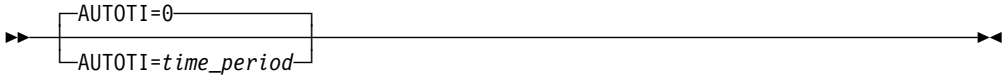
AUTORTRY=AUTOCAP
specifies that automatic logon requests are retried only when an adjacent CDRM or an adjacent CP which supports automatic logon is activated.

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



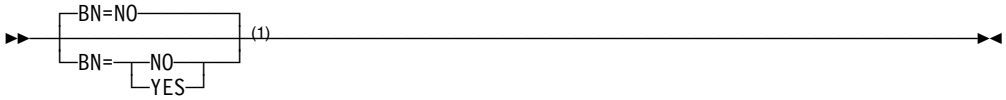
range: 0–7 days (or equivalent seconds, minutes, or hours)

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 requests are not retried.

You can change the value of AUTOTI with the MODIFY VTAMOPTS command while VTAM is running.

BN



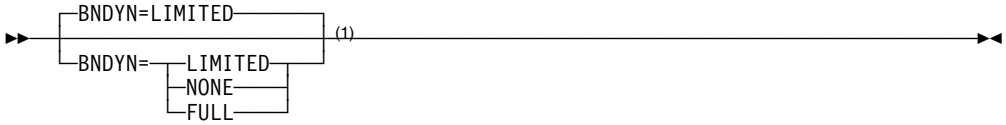
Note:
[1] BN is meaningful only if the NODETYPE=NN start option is also used.

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



Note:
[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.

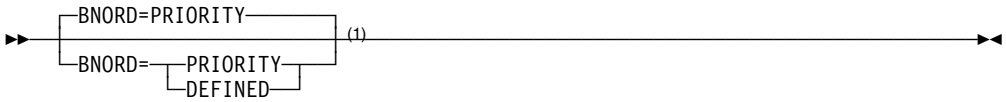
BNDYN=NONE
specifies that no entries will be added to the routing table except those explicitly identified by the NEXTCP definition statements.

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



Note:
[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.

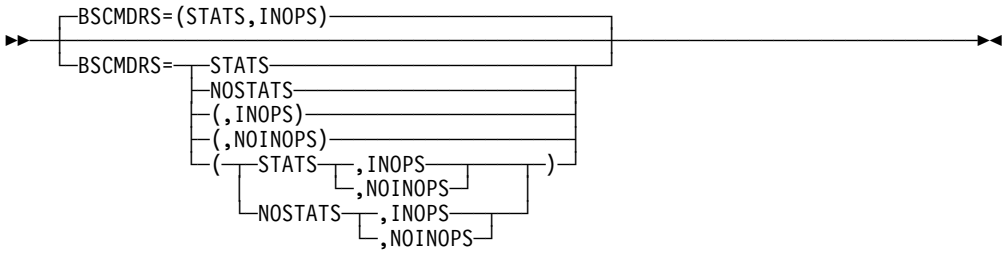
You can change the value of BNORD with the MODIFY VTAMOPTS command while VTAM is running.

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



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.

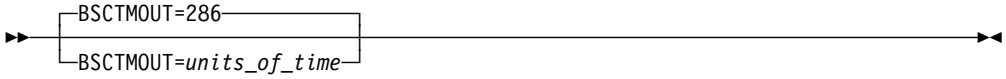
BSCMDRS=STATS
specifies that VTAM will record the BSC statistical MDRs in LOGREC.

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

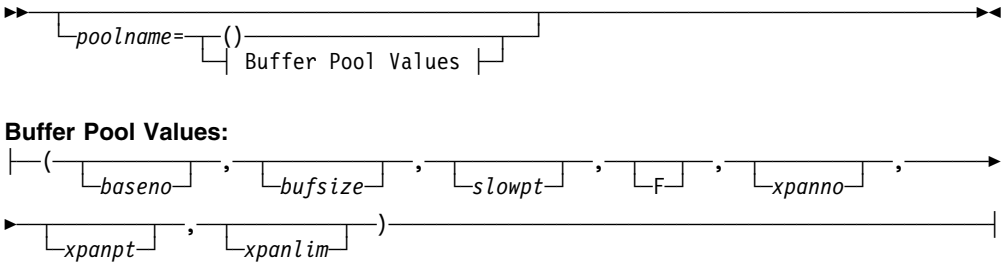


range: 0–2147483647

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



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 538 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 *VTAM Network Implementation Guide*.

poolname
is 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 65 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 *VTAM Network Implementation Guide*.

Table 65. VTAM Buffer Pools

Poolname	Pool ID	Abbreviation	Use
APBUF	AP00	AP [1]	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
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 (MVS) 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
XDBUF	XD00	XD	Provides storage for the I/O buffer during XID contact processing to peripheral nodes

Notes:

1. This buffer pool applies only to MVS.

baseno

range: 1–32767
dependency: *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: 64–4000 bytes
64–4016 bytes VM,VSE
dependency: *baseno*, IOBUF pool only LFBUF (VSE) only

specifies the number of bytes assigned to each buffer in the pool. If you specify a value less than 64, it is changed to 64.

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 539 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 539 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. If a channel-attached NCP is in this domain, the *bufsize* value for IOBUF pool must be equal to or greater than the value specified for the UNITSZ operand on the HOST definition statement in the NCP major node. Coding a value less than UNITSZ causes the NCP to not be activated.
5. Below are some resources that require minimum *bufsize* values. Refer to the appropriate device description manual for other resource types and additional information on the resources listed below.
   - For local SNA-attached peripheral nodes, the minimum *bufsize* is 182 bytes.

- For a LAN attached through the IBM 3172 Interconnect Controller, the minimum *bufsize* value for IOBUF is 182 bytes.
- For an multipath channel (MPC) connection, the IOBUF size must be a maximum of 15 bytes larger than the smallest WRITE device transmit buffer size. The transmit buffer size is a minimum of 4K.
- Some local SNA controllers, such as the 3174, require an even number and will not operate if *bufsize* is an odd value.

slowpt

dependency: *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. For VM, specifies that a buffer pool is locked in storage. 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 the *VTAM Network Implementation Guide* to determine which buffer pools can be pageable.

xpanno

dependency: *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 *VTAM 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 538).

VM VTAM can expand its buffer pools from any machine in the group that might require it. Consequently, if VTAM is to expand a locked buffer pool from any machine, then all the machines in the group require both the DIAG98 directory option and a virtual storage size large enough to completely contain the entire GCS shared system. Using this option does not cause VTAM to utilize any more locked pages then it would otherwise. If VTAM cannot expand the locked buffer pool from a particular machine in the group, an expansion of the buffer pool is scheduled in the VTAM machine.

Refer to the *VTAM Network Implementation Guide* for information and examples of dynamic buffer allocation.

xpanlim

dependency: IOBUF pool only, *xpanno*
range: 0–16384 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 66 on page 539 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 result.
     b. Round *bufsize* up to a doubleword (multiple of 8).
   - For IOBUF (no encryption specified):
     a. Add 87 bytes to the *bufsize* value.
     b. Round *bufsize* up to a doubleword (multiple of 8).
   - MVS For IOBUF, if YES, 24, or 31 is specified on the ENCRYPTN start option:
     a. Add 94 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.

Table 66. IBM-Supplied VTAM Buffer Pool Values

Buffer Pool Name	baseno	bufsize	slowpt	xpanno	xpanpt	xpanlim	minval
APBUF MVS [1]	16	56	2	1	3	-	-
BSBUF	18	216	0	1	17	-	-
CRPLBUF	60	160 MVS,VM 164 VSE	0	60	29	-	-
IOBUF	51	184 [2,3,4]	5	1	6	0	-
LFBUF	2	120 [4]	0	1	1	-	-
LPBUF	12	2032	0	6	2	-	4
SFBUF	51	112	0	1	1	-	5
SPBUF	2	144	0	1	1	-	-
XDBUF	6	681	0	1	5	-	-

Notes:

1. This buffer pool applies only to MVS.
2. VM,VSE For SDLC, LAN, X.25, and Local SNA connections to peripheral nodes, *bufsize* must be at least 182.
3. VM,VSE When accessing a LAN attached through the IBM 3172 Interconnect Controller, the minimum *bufsize* value for IOBUF is 182.
4. VM,VSE For SDLC lines attached through an integrated communication adapter, *bufsize* must be at least 204 bytes. For BSC lines attached through an integrated communication adapter, *bufsize* must be at least 285 bytes (286 bytes if the network also includes channel-attached SNA devices). For X.25 connections, *bufsize* is 196.

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 *VTAM 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, chose 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*.

To calculate how many buffers fit on a page (no encryption specified):

1. Determine the *bufsize* value.
2. Add 87 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.

The following example demonstrates this formula:

```
1. Determine the bufsize value                          279
2. Add 87 bytes                                       +  87
                                                      ------
                                                        366

3. Round the sum up to doubleword boundary              368

4. Divide 4096 by the value              4096 ÷ 368 =

5. Quotient                                          = 11 buffers
   Remainder                                         = 48 bytes
```

This example shows that each page of storage will contain 11 buffers and 48 unused bytes.

MVS If the encryption start option is specified (ENCRYPTN = YES, 24, or 31), add 94 to the *bufsize* value instead of 87.

Table 67 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.

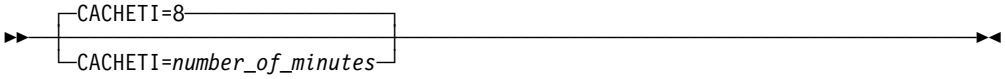
Table 67 (Page 1 of 2). IOBUF Buffer Size and Buffers per Page

Number of buffers per page	Range of bufsize — No Cryptography	Range of bufsize — Cryptography
1	1962–4000	1955–4000
2	1274–1961	1267–1954
3	938–1273	931–1256
4	730–937	723–930
5	594–729	587–722
6	498–593	491–586
7	426–497	419–490
8	362–425	355–418
9	322–361	315–354
10	282–321	275–314
11	250–281	243–274
12	226–249	219–242
13	202–225	195–218
14	186–201	179–194
15	170–185	163–178
16	154–169	147–162

Table 67 (Page 2 of 2). IOBUF Buffer Size and Buffers per Page

Number of buffers per page	Range of bufsize — No Cryptography	Range of bufsize — Cryptography
17	138–153	131–146
18	122–137	115–130
19	114–121	107–114
20	106–113	99–106
21	98–105	91–98
22	90–97	83–90
23	82–89	75–82
24	74–81	67–74
25	66–73	1–66
26	1–65	

CACHETI

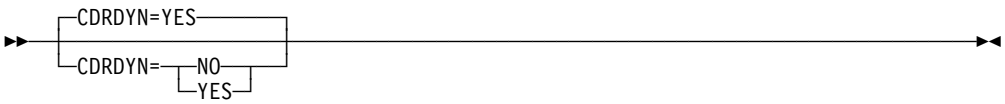


range: 0–1440 minutes

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



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.

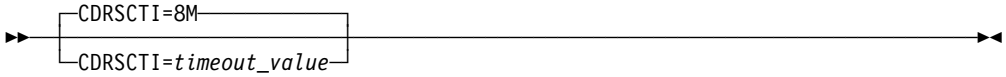
Notes:

1. For information specific to cross-domain sessions, see “CDRDYN” on page 151.
2. The CDRDYN start option value, or a MODIFY VTAMOPTS value, overrides the CDRDYN operand value on the 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



dependency: applies only to subarea resources
range: 1 second–7 days **MVS,VM**
1–32767 seconds (in S, M, H, or D) **VSE**

specifies the minimum amount of time for which the system retains a dynamically defined subarea 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 "CDRSCTI Start Option" in the *VTAM Network Implementation Guide* for details.

For a description of the retention time for APPN resources, see "DIRTIME" on page 550.

timeout_value can be specified in the following notation, where *n* is an integer:

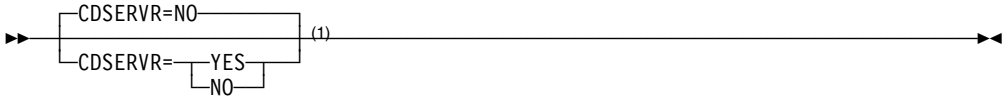
n or nS specifies the number of seconds a resource is retained

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



Note:
[1] CDSERVR is meaningful only if the NODETYPE=NN start option is also used.

specifies whether this network node is a central directory server. CDSERVR is valid only when NODETYPE=NN.

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.

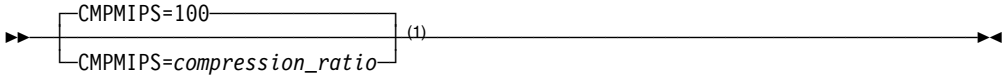
CINDXSIZ

Notes:
1 CINDXSIZ=8176 is the default for MVS and VM.
2 CINDXSIZ=4080 is the default for VSE.

range: 28–16777199 bytes

specifies the maximum size of the ISTCIT and ISTCONVT index tables.

CMPMIPS (MVS, VSE)



Note:
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.

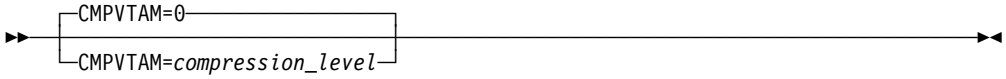
The higher the value specified, the greater the amount of time spent in adaptive mode. The lower the value, the sooner a compression table becomes static.

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



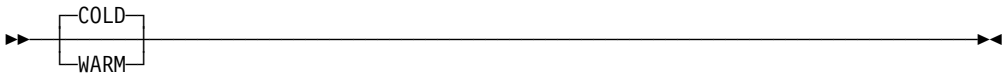
range: 0–4 **MVS,VSE**
0–1 **VM**

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 **MVS,VSE** Small adaptive compression table
3 **MVS,VSE** Medium adaptive compression table
4 **MVS,VSE** Large adaptive compression table

You can change the value of CMPVTAM with the MODIFY VTAMOPTS command while VTAM is running.

COLD and WARM (MVS, VSE)



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
```

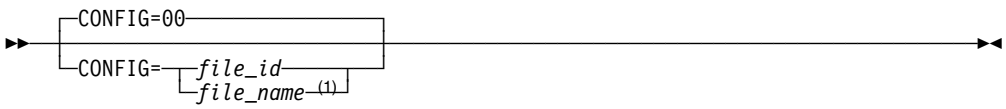
for each major node identified by the CONFIG start option.

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



Note:
[1] CONFIG= *file_name* applies to MVS and VSE only.

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 “Configuration Lists” in the *VTAM Network Implementation Guide*.

CONFIG=00
MVS,VM files a user-defined configuration list under the name ATCCON00.

VSE File a user-defined configuration list of major nodes under the name ATCCON00.B.

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.

MVS,VM File the list of major nodes under the name ATCCON*xx*.

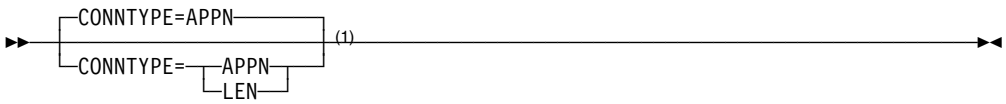
VSE File the list of major nodes under the name ATCCON *xx* .B. You can file more than one list of major nodes to give the user a choice of configurations and to avoid having to issue a separate VARY command to activate each major node. The operator, however, can choose only one configuration from the list.

CONFIG=*file_name* MVS,VSE
is 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.

MVS Include a DD statement using this data definition name in the VTAM start procedure. VSE Include DLBL and EXTENT statements for this file name in the VTAM start procedure.

CONNTYPE



Note:
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.

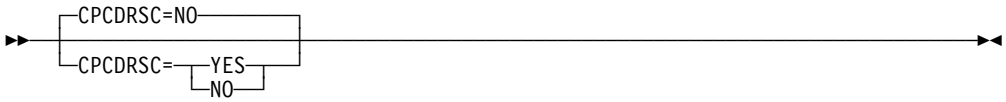
CONNTYPE=APPN
specifies that the connection is attempted as an APPN connection.

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

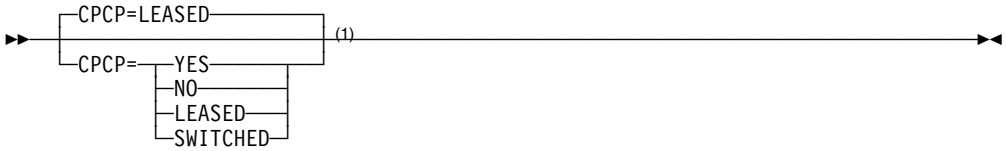


specifies whether VTAM allows applications or LUs to initiate and establish sessions to a dynamic LEN CP ILU. A dynamic LEN CP ILU is an ILU with a resource name that is the same as its adjacent link station CPNAME.

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



Note:
[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.

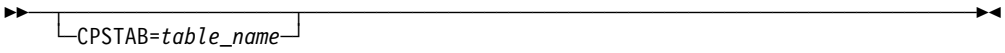
CPCP=YES
specifies that CP-CP sessions on all connections are supported.

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.

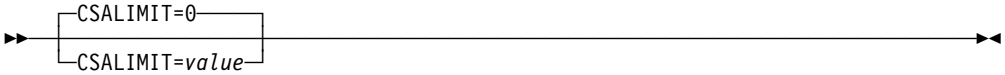
CPCP=SWITCHED
specifies that CP-CP sessions are supported for switched connections, but not supported for leased connections.

CPSTAB (VSE)



is the name of the call progress signal (CPS) table used to control retry values (limit and delay) for selected call progress signals received on X.21 SHM/MPS lines. *tablename* is the 8-character name of a CPS table.

CSALIMIT (MVS, VM)



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 these commands allows you to force the value specified to be used as the CSA usage limit.

Notes:

1. If the number you code is greater than the available CSA, no limit is enforced.
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.

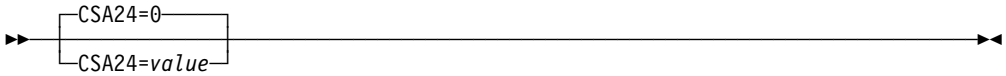
CSALIMIT=0
specifies that no limit is enforced on the amount of CSA used by VTAM.

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 (MVS, VM)



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.

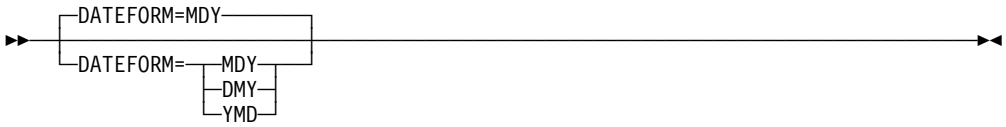
CSA24=0
specifies that no limit is enforced on the amount of 24-bit addressable CSA used by VTAM.

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



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.

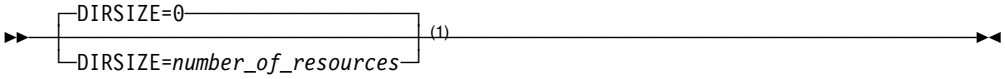
Note: You will encounter a MOSS clock problem if you set the IPL to occur after 15 days. For MOSS clock accuracy, the IPL should be less than or equal to 15 days. Otherwise, you can set the IPL up to 90 days in advance.

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 (-).

DIRSIZE



Note:
1 DIRSIZE is meaningful only if the NODETYPE=NN start option is also used.

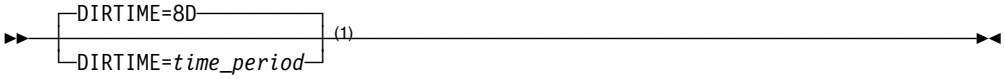
range: 0, or 1000–2147483647

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 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



Note:
1 DIRTIME is meaningful only if the NODETYPE=NN start option is also used.

dependency: applies only to APPN resources
range: 25 hours–2 years (730D)

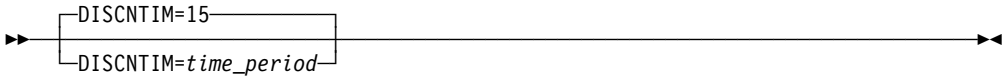
specifies the amount of time that an APPN resource can remain unused in the directory services database before VTAM deletes it. Resources older than the time specified are deleted.

You can specify the time in seconds (S), hours (H), or days (D).

DIRTIME is valid only when NODETYPE=NN, and applies only to APPN resources. See “CDRSCTI” on page 543 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



range: 1–255 seconds

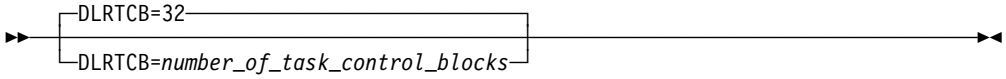
specifies the amount of time that VTAM delays deactivation of the SSCP-PU session when there are no outstanding LU-LU session requests.

You can specify the time in seconds (S).

DISCNTIM is valid only for PU types 2 and 2.1 that have DISCNT=DELAY specified on the PU definition statement.

You can change the value of DISCNTIM with the MODIFY VTAMOPTS command while VTAM is running.

DLRTCB (MVS, VM)



range: 1–32767

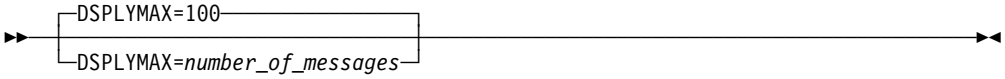
specifies the largest number of task control blocks (TCBs) used by VTAM for dump-load-restart subtasks and file I/O services. If you code 0, VTAM uses the default value of 32.

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.

DSPLYMAX



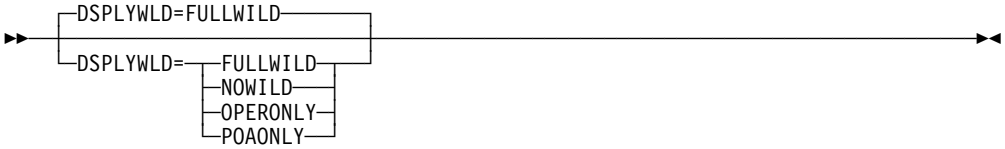
range: 1–65535

limits the number of messages displayed when the following commands are issued:

DISPLAY RSCLIST DISPLAY STORUSE

You can change the value of DSPLYMAX with the MODIFY VTAMOPTS command while VTAM is running.

DSPLYWLD



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.

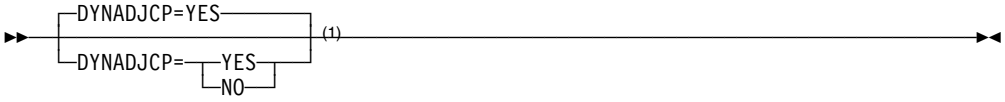
DSPLYWLD=FULLWILD
specifies that wildcards are permitted in DISPLAY commands from all network operators. Wildcards are permitted in DISPLAY commands from program operator applications whose APPL definition statement indicates DSPLYWLD=YES.

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.

DYNADJCP



Note:
[1] DYNADJCP is meaningful only if the NODETYPE start option is also used.

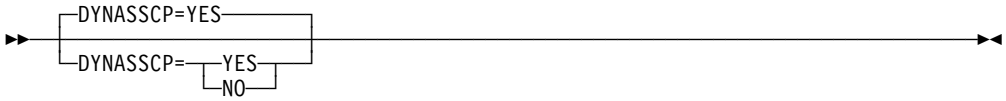
specifies whether adjacent control point (ADJCP) minor nodes are allowed to be created dynamically and placed in ISTADJCP. The ADJCP minor node provides control and management of connections between this node and an adjacent APPN type 2.1 node. This option can be overridden at the connection level (that 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



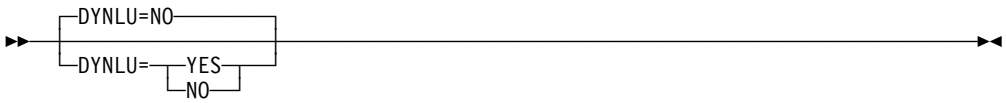
specifies whether VTAM dynamically routes session establishment requests to all active adjacent SSCPs if no appropriate adjacent SSCP table is defined.

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.

DYNLU



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.

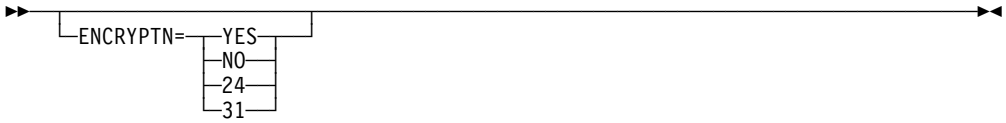
If you code DYNLU=NO, you are required to define all resources that can be accessed over adjacent link stations. If you do not define a given resource, sessions to or from that resource will fail.

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.

ENCRYPTN (MVS)



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:

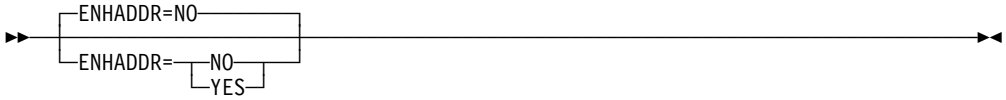
- If the currently active cryptographic product supports 24-bit addressing mode, VTAM supports cryptographic products that support 24-bit or 31-bit addressing mode and VTAM I/O buffers are placed in 24-bit storage.
- If the currently active cryptographic product supports 31-bit addressing mode, VTAM 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=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



specifies whether VTAM can assign element addresses greater than 65,535. These addresses are assigned to resources establishing sessions within this subarea.

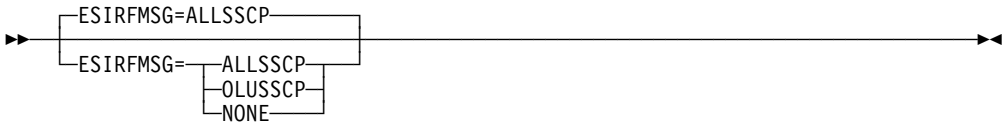
Note: If you specify ENHADDR=YES and have an application program that used the network-qualified address pair control vector X'15', refer to one of the following books:

- “Subarea/Element Address Pool Expansion” in the *VTAM Migration Guide for MVS/ESA*
- “Expanded Addressing Pool” in the *VTAM Migration Guide for VSE/ESA*
- “Expanded Addressing Pool” in the *VTAM Migration Guide for VM/ESA*.

ENHADDR=NO
specifies that you cannot use high-order element addresses for PLUs.

ENHADDR=YES
specifies that you can use high-order element addresses for PLUs.

ESIRFMSG



controls the display of messages IST891I, IST892I, and IST893I. These messages are issued when a session initiation request fails and extended sense data exists.

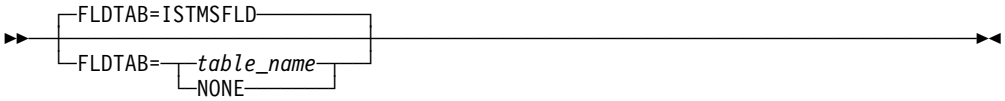
You can change the value of ESIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

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



specifies whether VTAM uses a message-flooding prevention table to suppress duplicate messages.

FLDTAB=ISTMSFLD
indicates that VTAM uses the internal message flooding table.

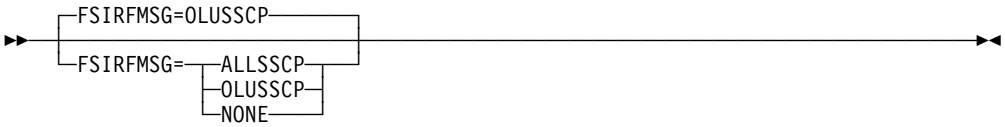
FLDTAB=NONE
indicates that VTAM will not suppress duplicate messages that are sent to the operator console or system hardcopy log.

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



controls the display of messages IST894I and IST895I. 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.

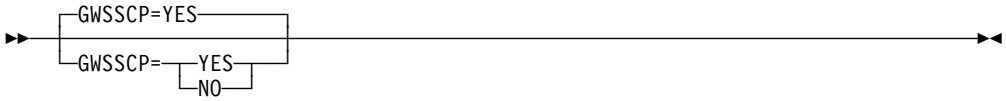
You can change the value of FSIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

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 (MVS, VM)



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:

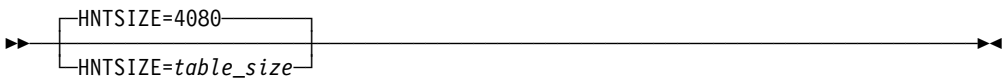
- Use the alias name translation facility and the alias selection function of the session management exit routine.
- Reroute initiation requests
- Perform gateway functions in a session setup path.

Note that GWSSCP=YES is meaningful only for APPN interchange nodes (NODETYPE=NN and HOSTSA is coded) and subarea nodes.

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 because it has no effect in this case.

If you coded a node to function as a migration data host (NODETYPE=EN and HOSTSA is coded), then you should code GWSSCP=NO. If you do not, the node will come up, but an error message will be issued.

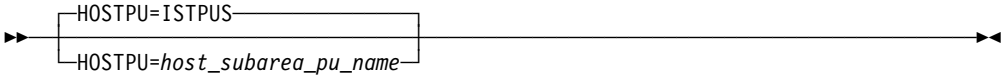
HNTSIZE



range: 4080–16777199 bytes

specifies the maximum size of the host node table used to find element addresses.

HOSTPU



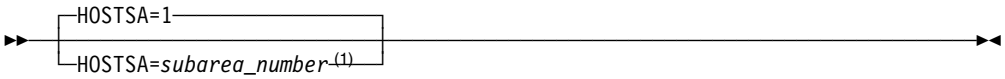
dependency: see notes

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



Note:
[1] HOSTSA provides subarea function. If you do not specify HOSTSA, but you specify NODETYPE, there is no default for HOSTSA and subarea function is not provided. If you do not specify HOSTSA and NODETYPE, a default of HOSTSA=1 is assumed and APPN function is not provided.

range: 1– to the maximum number of subareas allowed for the network

specifies the overall node configuration when combined with NODETYPE. When NODETYPE is not specified, HOSTSA defaults to 1. When NODETYPE is specified but HOSTSA is not specified, HOSTSA has no default and indicates that the node does not support subarea function. When HOSTSA is specified, the node continues to support subarea function regardless of whether NODETYPE is coded.

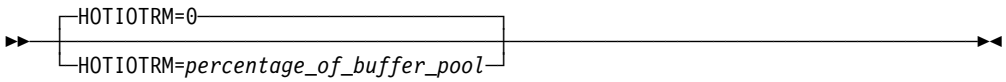
The combination of start options NODETYPE, HOSTSA and CPCP determine the overall node configuration and level of function support. The start options and resulting node configurations are intended to provide a smooth migration for existing subarea hosts.

Possible node configurations and their functional ability are summarized in Table 68 on page 569.

Note: If you have coded a node to function solely as an end node or network node (NODETYPE=EN or NN), then HOSTSA should not be coded.

HOSTSA must be less than or equal to the maximum number of subareas specified (MXSUBNUM start option). See “MXSUBNUM” on page 567 for more information on the MXSUBNUM start option.

HOTIOTRM



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.

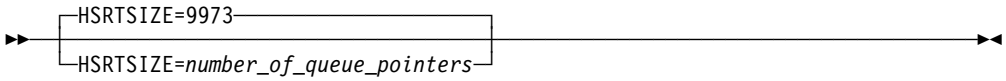
If you omit HOTIOTRM or specify 0, VTAM will not terminate sessions based on IO buffer pool usage.

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.

HSRTSIZE



range: 1–2097148

specifies the number of queue pointers in the symbol resolution table (SRT) for the network containing the VTAM host node.

INITDB

Note:
[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.

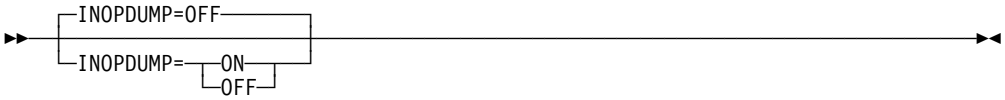
INITDB=ALL
specifies that both databases are loaded.

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 (MVS, VM)



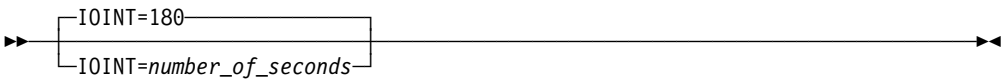
specifies whether VTAM dumps should be generated whenever VTAM detects certain channel-to-channel, local SNA, and channel-attached NCP INOPs.

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



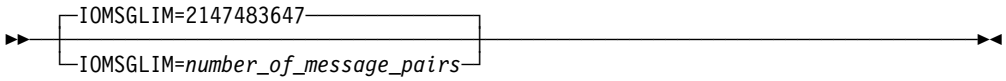
range: 0–5366000

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

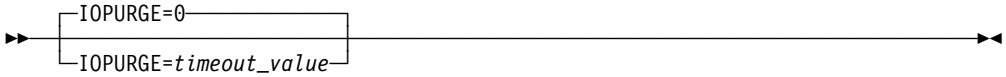


range: 1–2147483647

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



range: 30 seconds–7 days (or equivalent value in seconds, minutes, or hours)

specifies a time interval after which outstanding I/O requests are purged, and VTAM continues as if it had received negative responses from those requests. The types of outstanding I/O that are checked are CDINIT requests, direct search list requests, and APPN search requests.

Prior to the availability of this function (or when IOPURGE is set to zero), outstanding session requests could remain outstanding indefinitely. For example, when an intermediate host that is used in session routing goes down, no response is received to a session establishment request and the LU remains hung indefinitely. By setting an IOPURGE value, you can, in effect, cancel session requests that fail to complete in a certain amount of time. This frees the LU so you can request a session using an alternate path.

It is recommended that you set an IOPURGE value of three minutes or more. 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. A good rule of thumb is to set the IOPURGE value for twice as long as session setup usually takes. If you set IOPURGE too low, the session request will be canceled before a response can be received, and you will have to repeat the session request.

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.

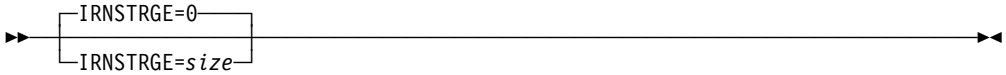
nM
specifies the number of minutes that can pass before a response is assumed to be lost.

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.

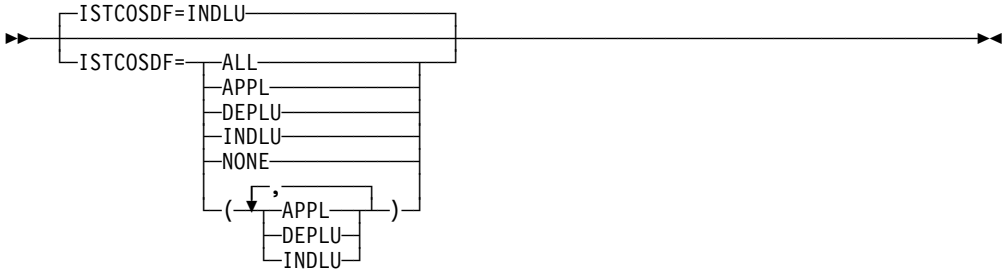
IRNSTRGE



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



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.

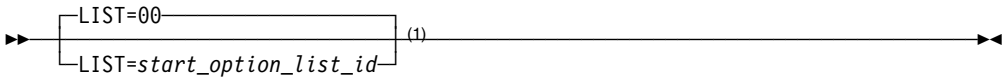
ISTCOSDF=APPL
indicates that ISTCOSDF is restricted to use by application programs.

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.

LIST

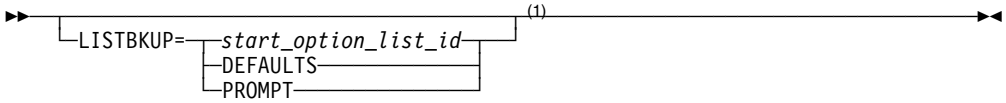


Note:
1 LIST can be entered by a VTAM operator only. If LIST is coded in an ATCSTR*xx* file, it is considered to be an error and is ignored.

specifies which list of predefined start options is used to initialize 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 “Creating Start Option Lists” in the *VTAM Network Implementation Guide*.

LISTBKUP



Note:
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.

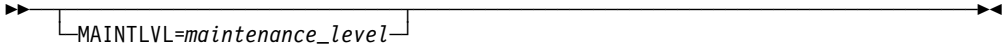
LISTBKUP=*start_option_list_id*
specifies the 2-letter start list identifier (ATCSTRyy) of another start file to process in place of the file in error. All options in the start file in error are set to the values before the file was processed and ATCSTRyy is processed.

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.

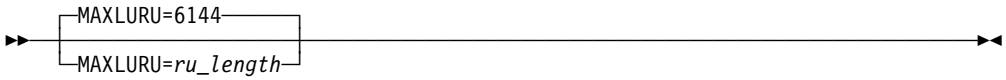
MAINTLVL



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.

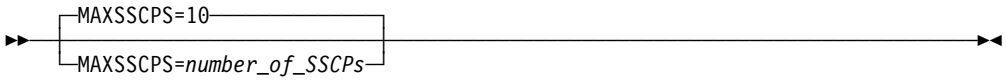
MAXLURU



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.

MAXSSCPS

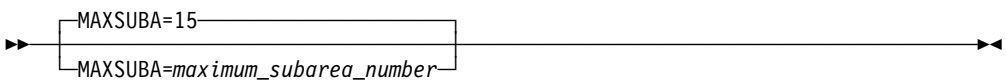


range: 1–255

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



dependency: 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.

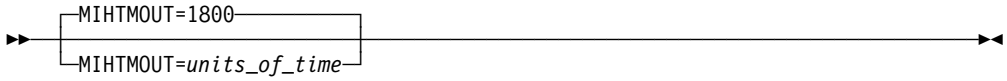
Note: If you have coded a node to function solely as an end node or network node (NODETYPE=EN or NN), then MAXSUBA should not be coded because it has no effect. If you code it, error messages will be issued, but the node will come up.

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 “How VTAM Handles Network and Subarea Addressing” and “Subarea Addressing” in the *VTAM Network Implementation Guide* for MAXSUBA usage considerations.

MIHTMOUT



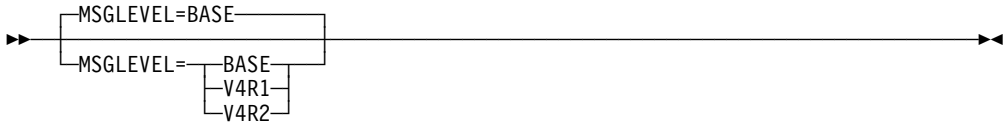
dependency: MVS does not apply to MPC
range: 1–2147483647

specifies the time (in tenths of a second) that allows you to adjust the missing interrupt handling time interval during channel-to-channel communication between two VTAMs or between VTAM and an IBM 3172 Interconnect Controller.

Note: MVS This start option does not apply to attachments between two hosts that use the multipath channel (MPC) feature. See“Multipath Channel Support” on page 58 and “Multipath Channel (MPC) Connections” in the *VTAM Network Implementation Guide* for information on MPC.

You can change the value of MIHTMOUT with the MODIFY VTAMOPTS command while VTAM is running.

MSGLEVEL



specifies the version of the messages to use for those messages listed in“Message Text for BASE and Version 4 Messages” in *VTAM Messages and Codes*. 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"Message Text for BASE and Version 4 Messages" in *VTAM Messages and Codes* 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.

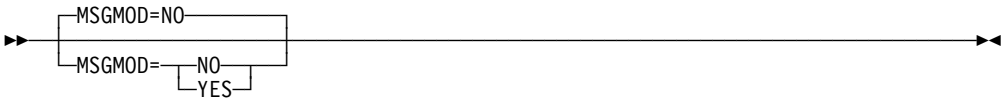
MSGLEVEL=V4R1
MVS specifies that VTAM issues the V4R1 message or messages, except for those that are coded otherwise in the USS table.

Refer to"Message Text for BASE and V4R1 Messages" in *VTAM Messages and Codes* 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"Message Text for BASE and V4R2 Messages" in *VTAM Messages and Codes* for a list of messages replaced by a V4R2 message or message group.

MSGMOD



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.

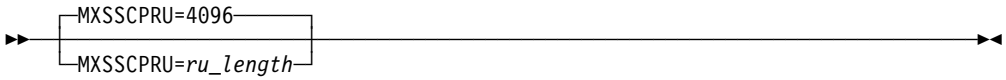
MXSAWBUF

Notes:
1 MXSAWBUF=10000 is the default for MVS and VM.
2 MXSAWBUF=500 is the default for VSE.

range: 1–2147483647

specifies the maximum number of buffers used for session awareness (SAW).

MXSSCPRU



range: 1–65535 bytes

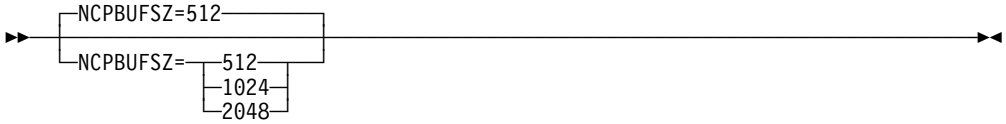
specifies the maximum length for RUs in SSCP sessions (SSCP-PU, SSCP-LU, and SSCP-SSCP).

MXSUBNUM

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



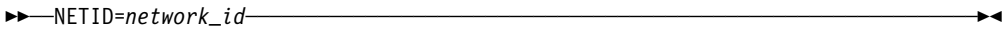
dependency: see notes

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

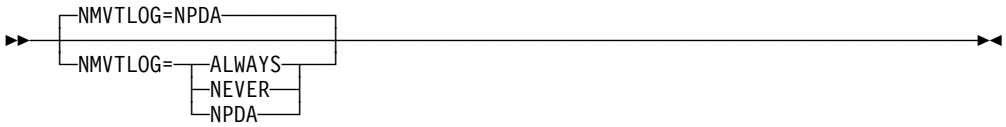


specifies the required 1–8 character identifier of the network containing the host.

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. In addition to NETID, a gateway (MVS, VM) configuration requires GWSSCP=YES and an SSCPNAME.

NMVTLOG



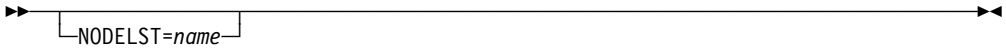
specifies when VTAM is to record network management vector transport (NMVT) records in LOGREC.

NMVTLOG=ALWAYS
specifies that VTAM always records NMVT alerts in LOGREC.

NMVTLOG=NEVER
specifies that VTAM never records NMVT alerts in LOGREC.

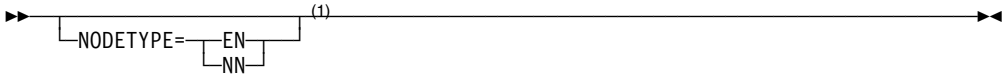
NMVTLOG=NPDA
specifies that VTAM records NMVT alerts in LOGREC only when the NetView hardware monitor, NPDA, is not active.

NODELST (MVS, VSE)



provides the 3–8 character name of the configuration restart file in which VTAM maintains a list of all currently active major nodes.

NODETYPE



Note:
[1] NODETYPE provides APPN function. The combination of NODETYPE and HOSTSA determines the configuration (subarea node, interchange node, migration data host, network node, or end node).

specifies the node type that is presented to other APPN nodes. This start option, together with the HOSTSA start option, specifies a specific node configuration. 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, HOSTSA is an optional start option and specifies whether the node will continue to support subarea function. Table 68 provides a summary of the functions that are implied by a given node configuration.

If a pure APPN node is desired, then the HOSTSA start option should not 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.

See Table 68 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 fail. However, if the node is a migration data host or an interchange node, FID4 connections are allowed.

Table 68. Node Type Functional Summary

Node Type	NODETYPE	HOSTSA	CP-CP Sessions	SSCP-SSCP Sessions	NCP Ownership	Interchange Function
Subarea Node	(not coded)	*n* [1]	no	yes	yes	no
Interchange Node	NN	*n* [1]	yes	yes	yes	yes
Migration Data Host	EN	*n* [1]	yes	yes	no [2]	no
Network Node	NN	(not coded)	yes	no	no [2]	no
End Node	EN	(not coded)	yes	no	no [2]	no

Notes:

1. *n* represents a subarea host number.
2. Activation of an NCP is not allowed.

NODETYPE=EN

specifies that the node is an end node. When NODETYPE=EN, the HOSTSA start option specifies whether the node continues to support subarea function. If HOSTSA is specified, the node operates as a migration data host. If HOSTSA is not specified, the node operates as an end node. When the node is

configured as a migration data host or end node, attempts to activate an NCP are failed. NCPs can be contacted, however.

NODETYPE=NN
specifies that the node is a network node. When NODETYPE=NN, the HOSTSA start option specifies whether the node continues to support subarea and interchange function as well. If HOSTSA is specified, the node operates as an interchange node. If HOSTSA is not specified, the node operates as a network node. Interchange nodes can own NCPs and their resources.

NOTRACE

For a description of the NOTRACE start option, see “TRACE for VTAM Internal Trace” on page 591 and “TRACE for Buffer, I/O, NCP Line, SIT, or SMS Traces” on page 588.

NQNMODE

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.

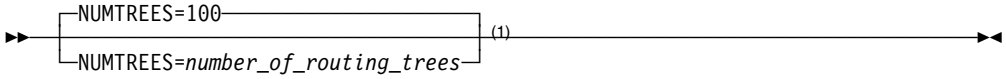
NQNMODE=NAME
specifies that the cross-network resource is defined by both its network-qualified name and its non-network-qualified name. The same-network CDRSC is always considered to be defined using NQNMODE=NAME because it is always identified by its non-network-qualified name. This is the mode in 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.

NUMTREES



Note:
[1] NUMTREES is meaningful only if the NODETYPE=NN start option is also used.

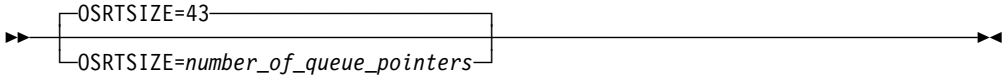
range: 2–10000

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.

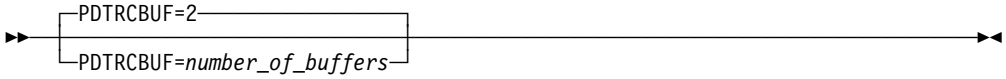
OSRTSIZE



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

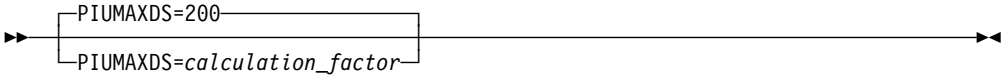


range: 2–255

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 (MVS)

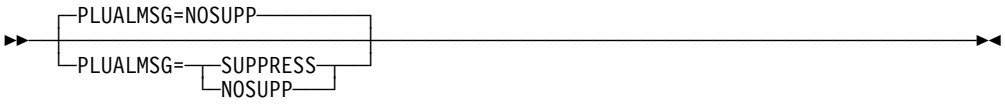


range: 0–2147483647

is 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



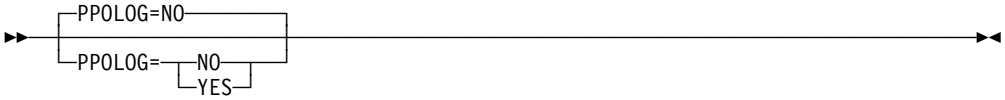
specifies whether VTAM is to suppress messages for a session setup failure caused when the PLU is not available for a session.

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 the session setup failure messages.

PPOLOG



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.

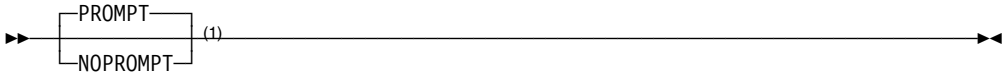
You can change the value of PPOLOG with the MODIFY VTAMOPTS command or the MODIFY PPOLOG command while VTAM is running.

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.

PROMPT and NOPROMPT



Note:
[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.

PSSTRACE (MVS)

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.

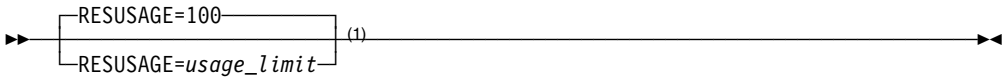
PSSTRACE=IRB
includes IRB entries in the VTAM internal trace.

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.

RESUSAGE



Note:
1 RESUSAGE is meaningful only if the NODETYPE=NN start option is also used.

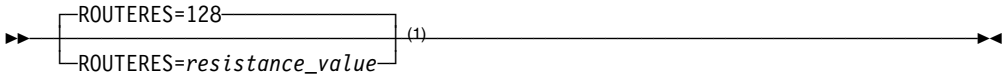
range: 0–2147483647

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



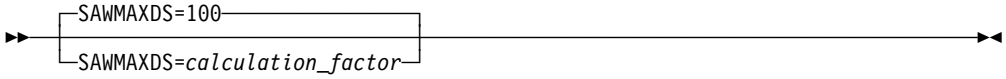
Note:
1 ROUTERES is meaningful only if the NODETYPE=NN start option is also used.

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.

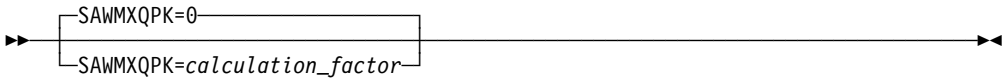
SAWMAXDS (MVS)



range: 1–2147483647

is a factor used in an equation to calculate the maximum buffer limit for SAW data space.

SAWMXQPK (MVS)

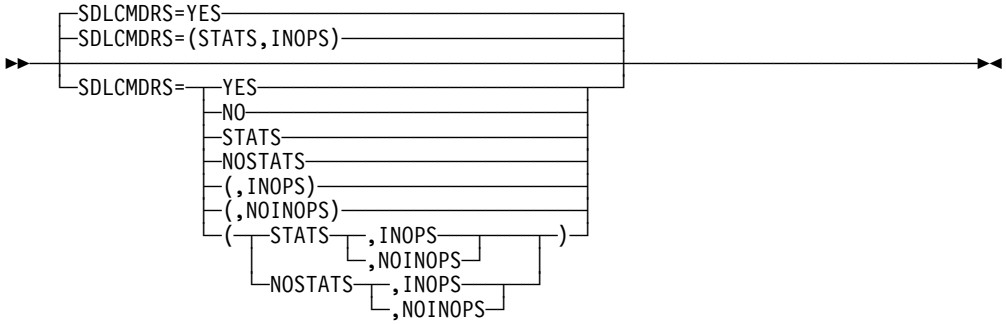


range: 0–2147483647

is 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



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.

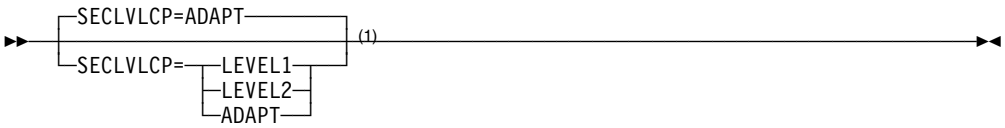
YES
STATS
specifies that VTAM records the SDLC statistical MDRs in LOGREC.

NO
NOSTATS
specifies that VTAM does not record the SDLC statistical MDRs in LOGREC.

INOPS
specifies that VTAM records the SDLC inoperative MDRs in LOGREC.

NOINOPS
specifies that VTAM does not record SDLC inoperative MDRs in LOGREC.

SECLVLCP (MVS, VM)



Note:
[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 a control point.

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 LU 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.

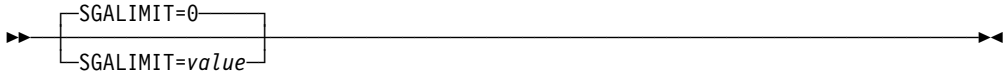
SECLVLCP=LEVEL2
specifies that the enhanced protocol is used. If a partner LU does not support the enhanced protocol, VTAM rejects the session and issues a sense code.

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 LU. VTAM attempts to use the enhanced protocol but will use the basic protocol if a partner LU 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.

SGALIMIT (VSE)



dependency: see notes
range: 0–2048M (2 gigabytes)

is the maximum amount of system GETVIS area (SGA) that can be used by VTAM. This limit is expressed as a number followed by a K (representing 1024 bytes) or an M (representing 1048576 bytes). If you do not code M or K, K is assumed.

You can change the value of SGALIMIT with the MODIFY VTAMOPTS command or the MODIFY SGALIMIT command while VTAM is running. The F operand of these commands allows you to force the value specified to be used as the SGA usage limit.

Notes:

1. If the number you code is greater than the available SGA, no limit is enforced.
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 SGALIMIT, define LPBUF so that it does not have to expand. Other possible consequences are lost messages or failures when sessions are initiated or terminated.

SGALIMIT=0
If you code 0 or let SGALIMIT default, no limit is enforced on the amount of SGA used by VTAM.

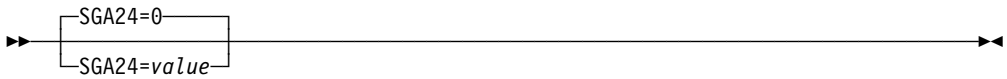
SGALIMIT=n or nK
n is the number of 1-kilobyte increments, in decimal, that is the maximum amount of SGA that can be used by VTAM. The value of *n* is rounded up to the next multiple of 4.

SGALIMIT=nM
n is the number of 1-megabyte increments, in decimal, that is the maximum amount of SGA that can be used by VTAM.

If neither M nor K is coded, K is assumed.

For more information on the system GETVIS area (SGA), refer to the *VTAM Network Implementation Guide*.

SGA24 (VSE)



range: 0–16M

specifies the maximum amount of 24-bit addressable system GETVIS area (SGA) 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 SGA addressable storage below the 16 MB line (24-bit addressable storage), the SGA24 limitations do not apply.

You can change the value of SGA24 with the MODIFY VTAMOPTS command or the MODIFY SGALIMIT command while VTAM is running. The F operand of these commands allows you to force the value specified to be used as the SGA usage limit.

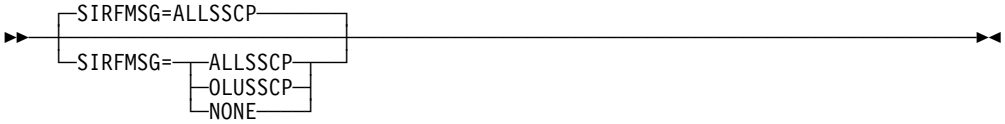
SGA24=0
specifies that no limit is enforced on the amount of 24-bit addressable SGA used by VTAM.

SGA24=value
specifies the maximum amount of SGA 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 SGA 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 SGA that can be used by VTAM.

SIRFMSG



controls the display of IST663I, IST664I, IST889I, and their message groups. These messages are issued when a session initiation request fails.

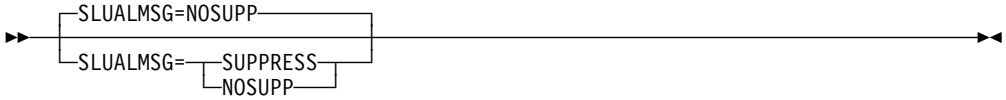
You can change the value of SIRFMSG with the MODIFY VTAMOPTS command while VTAM is running.

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.

SLUALMSG



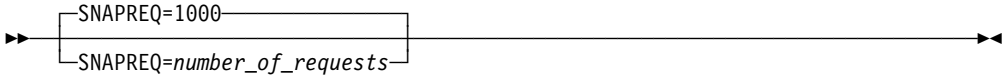
specifies whether VTAM is to suppress messages for a session setup failure caused when the SLU is not enabled for a session.

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.

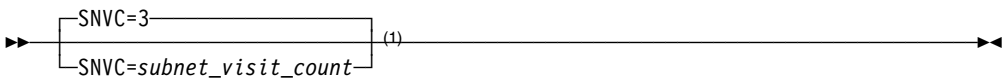
SNAPREQ



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



Note:
[1] SNVC is meaningful only if the BN=YES start option is also used.

range: 1–255

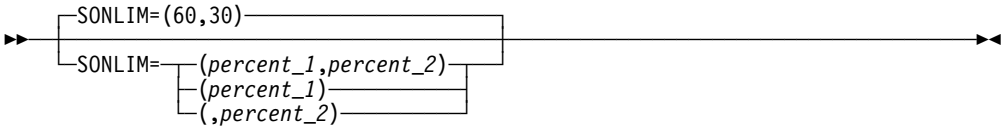
defines a maximum number of networks this border node will search when looking for a resource. This start option is valid only when BN=YES for this node.

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



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.

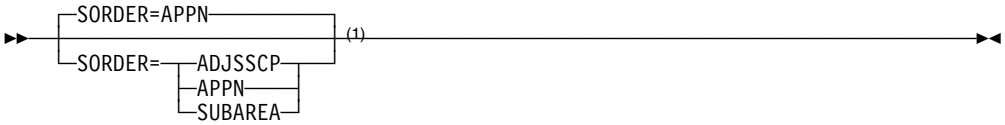
The session outage notification does not pre-allocate buffers. No I/O buffers are allocated until SON processing begins.

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



Note:
[1] SORDER is meaningful only in a network node, interchange node, or 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 is overridden if the target LU's location is already identified and SSCPORD=PRIORITY is specified. SORDER is valid only for network nodes (NODETYPE=NN), and end nodes functioning as migration data hosts.

You can change the value of SORDER with the MODIFY VTAMOPTS command while VTAM is running.

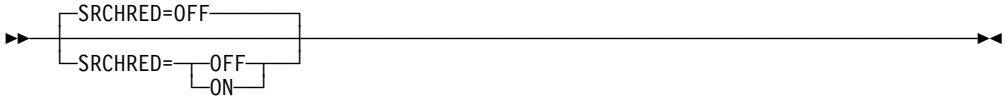
See “cdrmname” on page 488 for related information on network searches.

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 the APPN network is to be searched first, regardless of whether (or where) the operand ISTAPNCP is coded in the adjacent SSCP table.

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



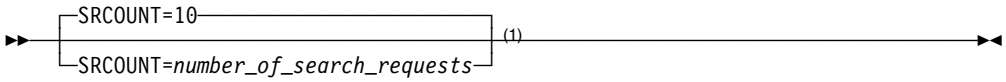
specifies whether this node can reduce searches for resources which are found to be unreachable. Refer to “Improving VTAM Performance Using Start Options” in the *VTAM Network Implementation Guide* for information about using the search reduction facility.

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



Note:
[1] SRCOUNT is meaningful only if the SRCHRED=ON start option is also used.

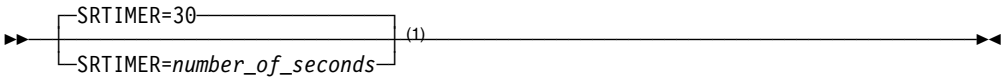
range: 0–65535

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



Note:
[1] SRTIMER is meaningful only if the SRCHRED=ON start option is also used.

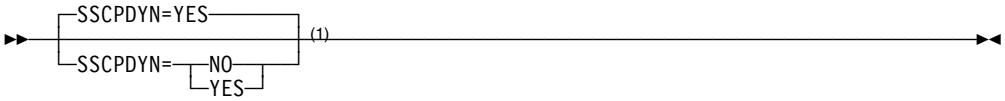
range: 0–65535 seconds

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



Note:
[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.

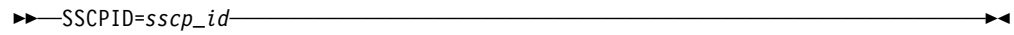
If you specify SSCPDYN=NO, you might have to define additional adjacent SSCP tables. When an application issues a CLSDST PASS macroinstruction during session setup, VTAM does not build an adjacent SSCP table in the application's network. (When SSCPDYN=YES is coded in the application's network, VTAM builds such a table automatically.)

This start option is closely related to SSCPORD. Refer to “Improving Performance” in the *VTAM 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



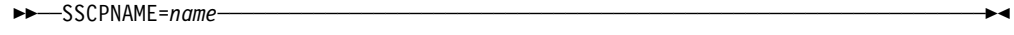
range: 0–65535

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.

Note: 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.

SSCPNAME



dependency: see notes

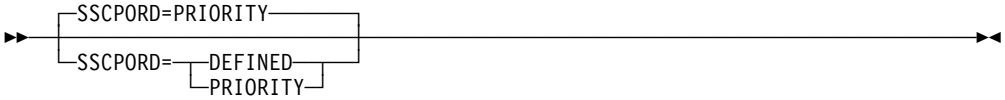
specifies the required 1–8 character name of the VTAM SSCP or CP. You should use the same name here that you use on the host CDRM in multiple-domain networks. This start option is required.

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 “Type 2.1 Casual Connection” in the *VTAM Network Implementation Guide* for information about casual connection.
- The SSCPNAME should match the name on the host CDRM definition statement. See “cdrmname” on page 151 for further details.

SSCPORD



specifies whether VTAM searches an adjacent SSCP table in priority order (the default) or in the order in which the table is defined.

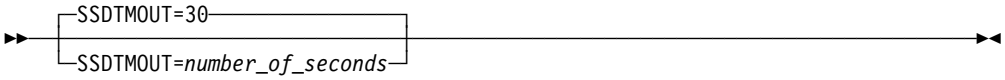
This start option is closely related to SSCPDYN. Refer to “Improving Performance” in the *VTAM Network Implementation Guide* for information on how to use these two start options.

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

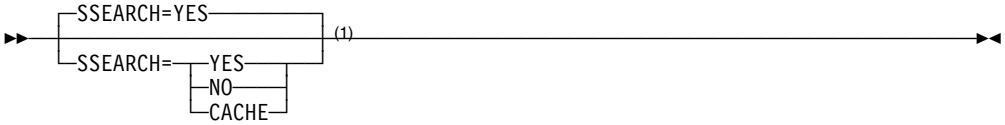


range: 1–65535

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



Note:
[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.

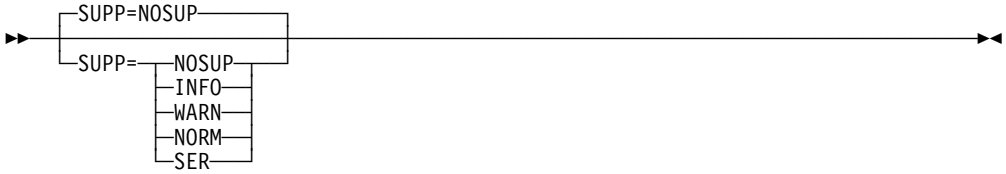
You can change the value of SSEARCH with the MODIFY VTAMOPTS command while VTAM is running.

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=NO
specifies that the subarea network is not to be searched.

SSEARCH=YES
specifies that the subarea network is to be searched.

SUPP



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.

MVS 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. 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 “Message Suppression Levels” in *VTAM Messages and Codes* for individual suppression levels.

Note: Messages that are suppressed will be recorded in the PPO log, even though they will not appear on the system console.

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.

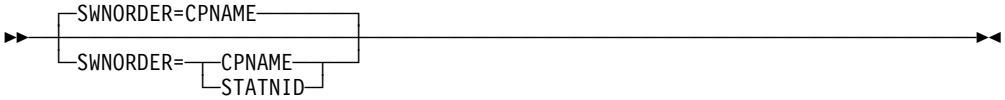
SUPP=WARN
specifies that warning messages (as well as informational messages) are suppressed. Warning messages identify error conditions that do not cause commands to fail or be rejected. These messages inform the operator that there is a problem, such as an operand that is not valid or a minor node 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



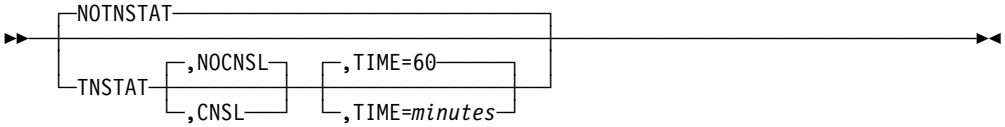
specifies the way VTAM locates a switched PU. If you do not code SWNORDER, VTAM searches for a PU by the CPNAME first. If VTAM does not find the PU by the CPNAME, it searches by the station identifier (IDBLK and IDNUM operands on the PU definition statement for the switched major nodes).

You can change the value of SWNORDER with the MODIFY VTAMOPTS command while VTAM is running.

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.

TNSTAT



specifies whether VTAM keeps tuning statistics. Refer to Chapter 11, “Tuning VTAM for Your Environment” in the *VTAM 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.

MVS 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:

MVS SMF data set
VM "FILE TUNSTATS A" CMS file.
VSE Trace file (an external file to which SYS001 has been assigned).

NOCNSL
specifies that the tuning statistics records are written only to the:

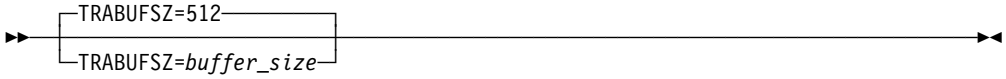
MVS SMF data set
VM "FILE TUNSTATS A" CMS file
VSE Trace file.

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.

Note: VM You can direct trace output to any file by issuing a FILEDEF command before you start VTAM. The default file name is FILE TUNSTATS.

TRABUFSZ (VM, VSE)

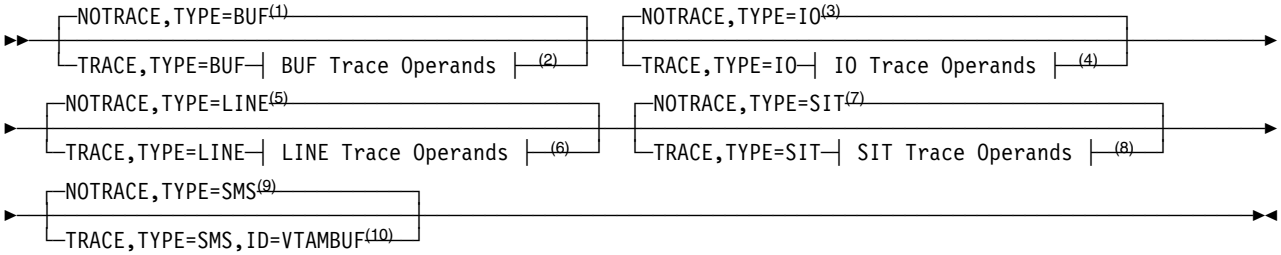


range: 96–65535 (multiples of 8)

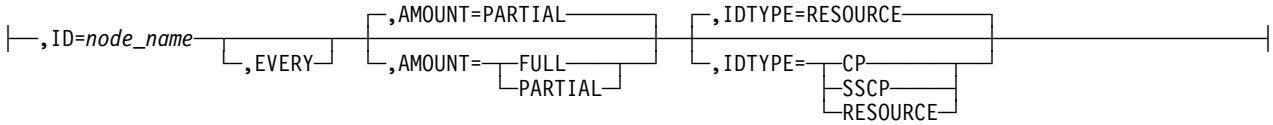
sets the buffer size, in bytes, used in the I/O adapter for token-ring connection.

If the value is not a multiple of eight, VTAM will not accept TRABUFSZ for processing.

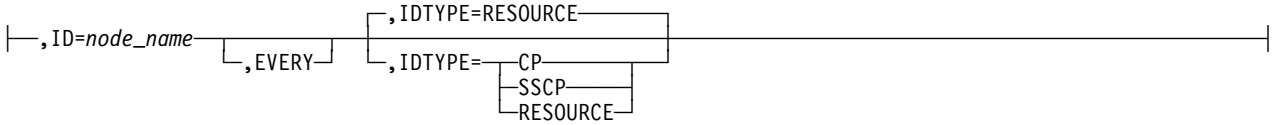
TRACE for Buffer, I/O, NCP Line, SIT, or SMS Traces



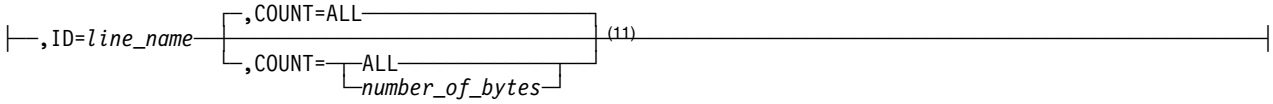
BUF Trace Operands:



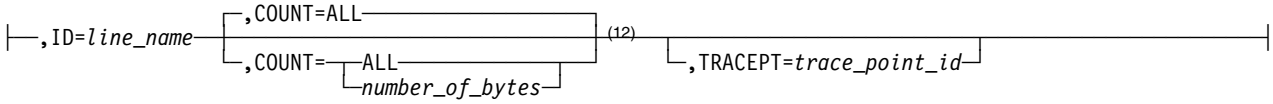
IO Trace Operands:



LINE Trace Operands:



SIT Trace Operands:



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 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

4 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

5 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

6 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

7 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

8 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

9 Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list.

10 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

11 COUNT applies only to the IBM 3720 and 3745 Communication Controllers.

12 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**
is 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.

Note: 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.

Note: IDTYPE is valid only for the buffer contents and I/O 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. See the description of the MODIFY TRACE command in *VTAM Operation* for a list of the types of nodes you can choose.

Note: 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 ID and TYPE operands. 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.

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 Diagnosis Guide*.

TYPE=BUF
specifies whether a VTAM buffer trace is started or stopped.

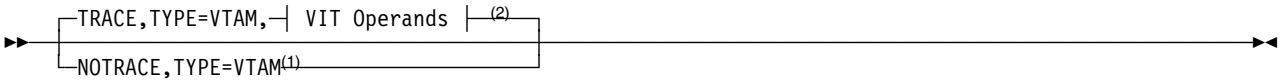
TYPE=IO
specifies whether a VTAM I/O trace is started or stopped.

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.

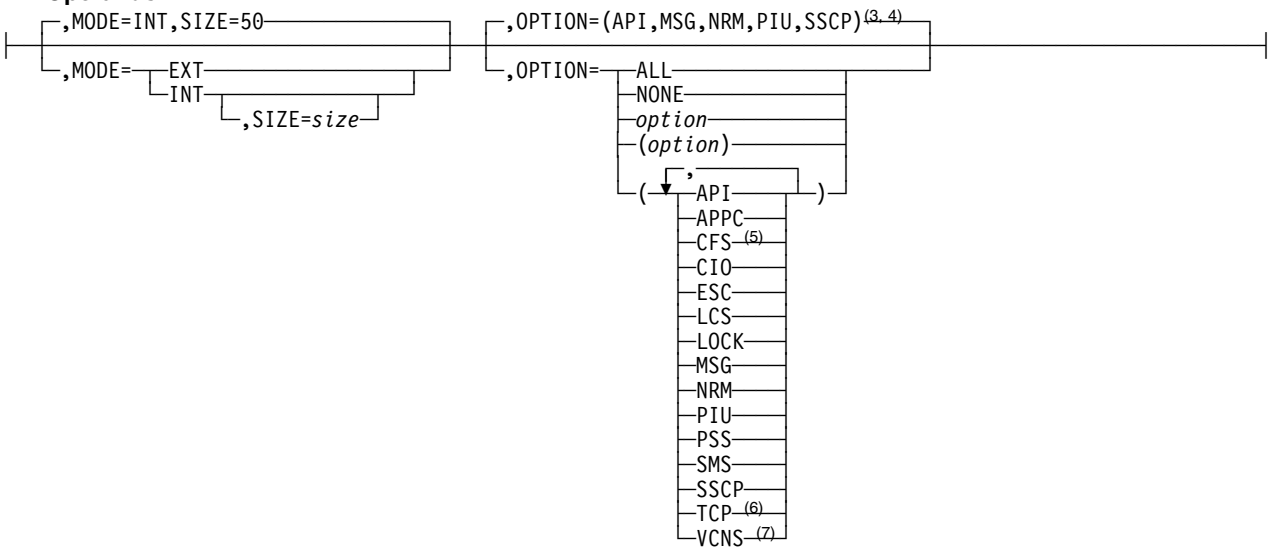
TYPE=SIT
specifies that a scanner interface trace is started or stopped for a 3720, 3725, or 3745 communication controller. There can be up to 8 concurrent line traces and SITs for each NCP, depending on how each NCP was generated.

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.

TRACE for VTAM Internal Trace



VIT Operands:



Notes:

1 NOTRACE,TYPE=VTAM is accepted but ignored. Tracing is started with the default trace table size and the default options.

2 Code TRACE and its qualifiers on one line. Code the TYPE qualifier immediately following TRACE.

3 The default options apply only to MODE=INT for MVS and VM.

4 The default options apply to MODE=INT and MODE=EXT for VSE.

5 The CFS trace option applies to MVS only.

6 The TCP trace option applies to MVS only.

7 The VCNS trace option applies to MVS and VM only.

specifies the options on the VTAM internal trace (VIT).

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.

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).

Note: MODE applies only to TYPE=VTAM.

MVS,VM 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.

VSE You can record trace data internally or externally, but not at the same time. VTAM always runs with the default trace options recorded, regardless of whether you request tracing. The default options are recorded internally or externally, depending on how the trace is started.

VSE If trace options are being recorded in either mode, and you issue a MODIFY TRACE command with a different mode, the current options continue to be recorded in the new mode.

You must run specific operating system utilities to trap, format, and view external trace output. SeeChapter 9, “Using the VTAM Internal Trace (VIT)” in *VTAM Diagnosis* for more information about use of these operating system utilities.

MVS,VM 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.

MVS 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.

VSE You can run the VTAM trace print utility (TPRINT) as a subtask under VTAM or as a job step under VSE to format the output. Refer to the MODIFY SUBTASK command in *VTAM Operation* for instructions on how to attach TPRINT.

The trace information is written sequentially into a tape or direct-access file. The file must be defined in DLBL and EXTENT statements, with the file name TRFILE as the first operand of the DLBL statement. When the trace file is full, the file is overwritten, with the newest trace records overlaying the oldest trace records. Subsequently, the trace-print facility can be used to write the formatted trace information onto SYSLST. The trace records are written to TRFILE in 2048-byte blocks. The number of tracks of direct-access storage allocated depends on the number of 2048-byte

records that you want to save. The VTAM trace facility is described in *VTAM Diagnosis*

VM You can issue the CPTRAP TRSOURCE command to enable the recording of GCS and VTAM virtual machine trace data in the CPTRAP TRSOURCE spool file. The VM TRACERED service routine formats trace data in the CPTRAP TRSOURCE spool file using a VTAM exit routine. The trace analysis program (TAP), which is part of the System Support Programs (SSP), provides extended formatting for VTAM and NCP trace records.

Refer to "Formatting and Printing Trace Records" in *VTAM Diagnosis* for a more detailed description of formatting and printing trace output.

NOTRACE,TYPE=VTAM
is accepted but ignored. Tracing is started with the default trace table size and the default options.

OPTION=(option,option,...option)
is a listing of the VTAM functions that you want to trace. This operand can be abbreviated as OPT.

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.

You can code one or more of the following options. However, if you code ALL or NONE, you cannot code any other options.

Note that each trace option can affect the performance of your host and the VTAM network. Be careful when you specify OPTION=ALL until you understand the effect this function can have on performance in your environment.

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. MVS,VM 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
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.

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 APPCCMD macro invocations, user posts and

exit scheduling by LU 6.2 code, calls to a security manager for security processing, and message unit transmissions between LU 6.2 components.

CFS MVS
helps you assess and manage your coupling facility structure.

CIO
helps you isolate problems related to channel I/O. CIO entries are written for attentions, error recovery, interruptions, HALT I/O SVC, and START I/O SVC.

ESC
helps you track, in detail, the flow of requests for a given process.

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 MVS
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 MVS,VM
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.

SIZE=*size*
is the number of pages in the internal trace table. (VSE SIZE has another meaning for external tracing. See the following special considerations for VSE.) SIZE is valid only when MODE=INT is specified. *size* is a decimal integer in the range 1–999. If you specify a value less than 50, 50 will be 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.

VSE The following considerations apply:

When recording externally, the value specified for the SIZE operand on the MODIFY TRACE command or the TRACE start option is used in a formula to determine the number of 4K fixed internal buffers that can be placed in the area allocated for the trace tables. The formula follows:

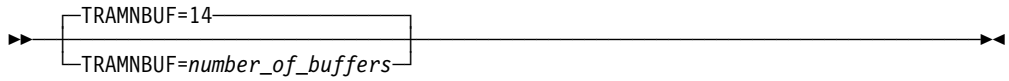
```
((SIZE x ATCPGSIZE) - 16) / 2K
```

The 16 bytes that are subtracted represent the buffer header, where no data can be placed.

If the SIZE operand is changed while MODE=EXT is in effect, the current table contents are written to the trace file before the old table is freed.

Note: If the requested size is not available, VTAM returns to the default value of SIZE=50.

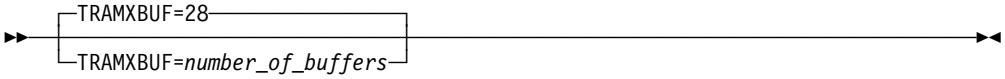
TRAMNBUF (VM, VSE)



range: 0–255

is the minimum number of buffers that are used as transmit buffers in the I/O adapter for token-ring connections.

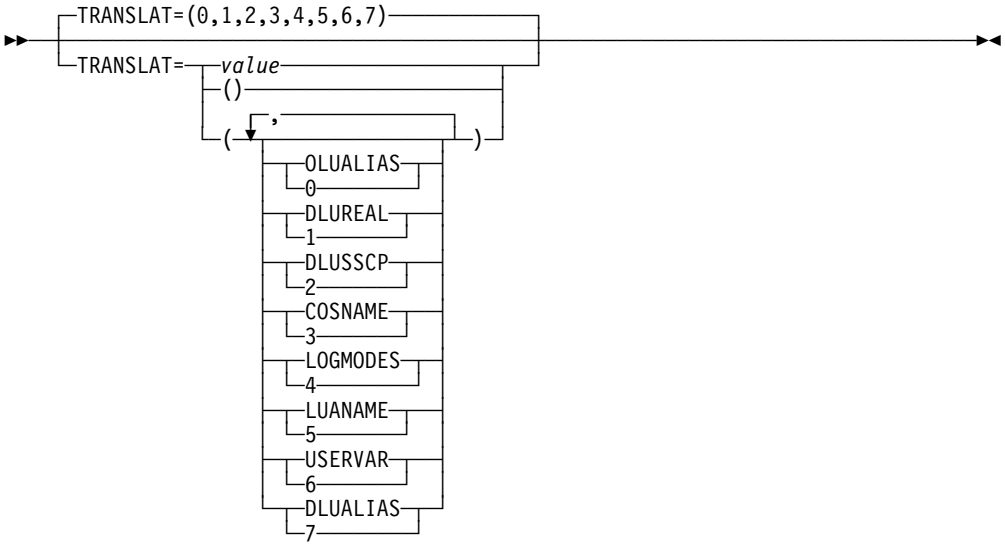
TRAMXBUF (VM, VSE)



range: 1–255

is the maximum number of buffers that are used as transmit buffers in the I/O adapter for token-ring connections.

TRANSLAT



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.

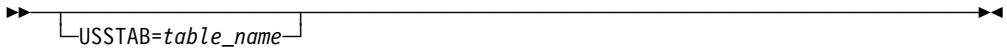
TRANSLAT=LOGMODES or 4 (bit 4)
specifies that VTAM is to determine logon mode names.

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.

USSTAB

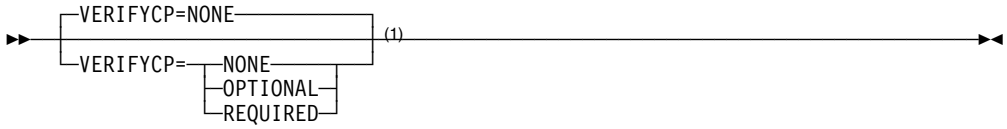


specifies the name of a USS table to be used for VTAM operator messages and for VTAM operator commands that are processed through USS. *table_name* is the 8-character name of a USS table. Choose one of the following USS tables:

- 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 (MVS, VM)



Note:
[1] The VERIFYCP start option is meaningful only if the NODETYPE start option is also used.

specifies whether VTAM performs LU-LU 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.

VERIFYCP is valid only when NODETYPE is specified.

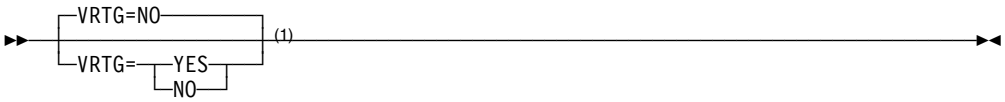
If you code VERIFYCP=OPTIONAL or VERIFYCP=REQUIRED, a security management product, such as RACF 1.9 or later, must be installed and active.

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

VERIFYCP=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 “LU 6.2 Security” in the *VTAM Network Implementation Guide* for information about how VTAM determines when to perform LU-LU verification.

VERIFYCP=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.

VRTG



Note:
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.

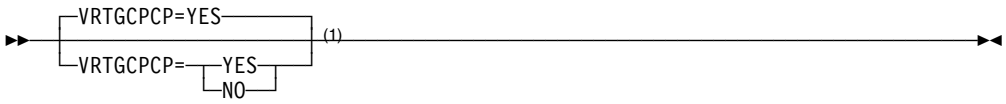
Note: This value can be overridden by coding the VRTG operand in the CDRM definition statement.

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



Note:
[1] VRTGCPCP is meaningful only if the NODETYPE and HOSTSA start options are also used.

dependency: 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.

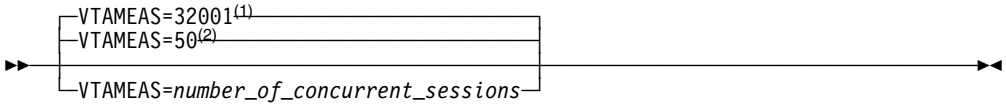
Note: This value can be overridden by coding the VRTGCPCP operand in the CDRM definition statement.

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



Notes:
1 VTAMEAS=32001 is the default for MVS and VM.
2 VTAMEAS=50 is the default for VSE.

range: 0–2147483647

specifies the number of concurrent sessions VTAM can have with other LUs.

XNETALS

Note:
1 The default for XNETALS is NO if BN=NO. The default for XNETALS is YES if BN=YES.

dependency: 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.

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.

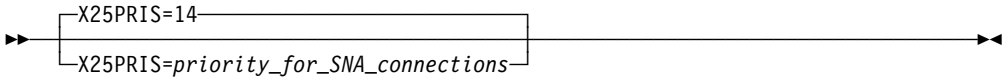
During LU-LU session establishment:

- 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 disallow nonnative network attachment for LU-LU sessions for a particular PU by coding NETID=(,NOXNETLS) on that PU. When NETID=(,NOXNETLS), VTAM indicates to the NCP that this PU will use VTAM's native network ID.

See the description of the NETID operand in the NCP and switched major node sections for details.

X25PRIS (VM)



range: 0–14

specifies a connection priority for SNA-X.25 connections, according to the following values:

0–3 low priority

4–7 normal priority

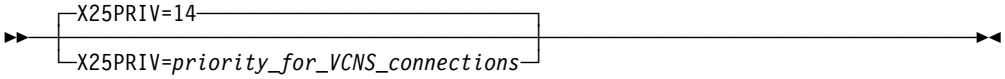
8–11 high priority

12–14 extra-high priority

The value you specify is used unconditionally for all SNA-X.25 connections.

Values which correspond to the same connection priority are treated as equivalent. For example, there is no effective difference between values 5 and 7. Both values are registered as the normal connection priority.

X25PRIV (VM)



range: 0–14

specifies a connection priority for VCNS (VTAM Common Network Services)-X.25 connections, according to the following values:

0–3 low priority

4–7 normal priority

8–11 high priority

12–14 extra-high priority

The value you specify is used for a given VCNS-X.25 connection only when no valid data priority value is negotiated when the connection is established.

Values which correspond to the same connection priority are treated as equivalent. For example, there is no effective difference between values 5 and 7. Both values are registered as the normal connection priority.

Chapter 5. VM/SNA Console Support (VSCS)

About This Chapter

The VM/SNA Console Support (VSCS) application program mediates between VTAM and a VM system. For instance, if end users in a VTAM network want to log on to VM, they must first log on to the VSCS application running in the VM host. However, VSCS can be made transparent to end users.

To provide SNA console support, VSCS uses:

- SNA console communications services (CCS), a part of VM, which provides full VM console interface capabilities for SNA terminal users
- The inter-user communication vehicle (IUCV), which transfers messages between VSCS and CCS.

This chapter describes:

- How to code VSCS start options on the DTIGEN macroinstruction
- How to install VSCS start options
- How to specify each VSCS start option
- How to code logon mode tables for VSCS devices.

VSCS Start Options

VSCS has its own set of start options. These options are coded as operands on a DTIGEN macroinstruction in a CMS file called DTIUSER*n*, where *n* is a number from 0–9.

Table 69 (Page 1 of 2). VSCS Start Options on the DTIGEN Macroinstruction

Name	Macroinstruction	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	DTIGEN	ACBLOOP	608			3	
		APPLID	608			VM	
		AQLIMIT	609			16384	
		BFRFIFO	609			N	could affect performance
		BLKMULT	610			1	
		CONFTXT *	610			N	
		CSTRACE	610			Y	could affect performance
		DEF3278 *	611			N	
		DEXIT *	611			N	
		DMPFREQ *	611			0	
		DMPINTC *	612			0	
		DMPSYSC *	612			0	
		DMPTIME	613			90	
		DPACE *	613			screen depth	
		DPTRACE	613			Y	
		DPXMTL *	614			1948	
		DTIUSER	614			0	
		DTTRACE	615			N	
		EXTRACE	615			N	could affect performance
		FRTRACE	615			N	could affect performance
		FSREAD *	616			Y	display terminals

Table 69 (Page 2 of 2). VSCS Start Options on the DTIGEN Macroinstruction

Name	Macroinstruction	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
		GETRACE	617			N	could affect performance
		IDENT	617			yyddd	
		ILUSIZE	617			40	
		ILUTRAC	618			N	
		KEXIT *	618			N	
		KPACE *	619			10	
		KPXMTL *	619			284	
		LEXIT	619			N	
		LGNCMDS	620				
		LURTRY *	620			5	
		ONELOGO *	620			N	
		OPTRACE	621			N	
		PASSWRD	621				
		PRNTNUM	621			16	
		PRTSHR *	622			Y	
		QDEPTH	622			32600	
		RCVBFRL *	622			284	
		RDSPTMR	623			10	
		RPLNUM	623			8	
		SCHED	624			N	
		SCIPCNT *	624			2	
		SCIPNUM *	625			0	
		SCIPTIM *	625			0	
		SNACCS	626			Y	
		SPEC	626			N	
		STCHKTM *	626			0	
		STRELTM *	627			0	
		STTRACE	627			N	
		TIMECPY *	627			3	
		TIMEREL *	628			120	
		TRASIZE	628			2000	
		TSKRTRY *	628			10	
		VSAMLM *	629			10	
		VTTRACE	629			Y	could affect performance
		W2741L *	629			129	
		W3767L *	630			129	
		WTWXL *	630			72	

Note:

* After VSCS is started, the operator can use the VSCS CHANGE command to dynamically change this operand. Refer to *VTAM Operation* for more information about the VSCS CHANGE command.

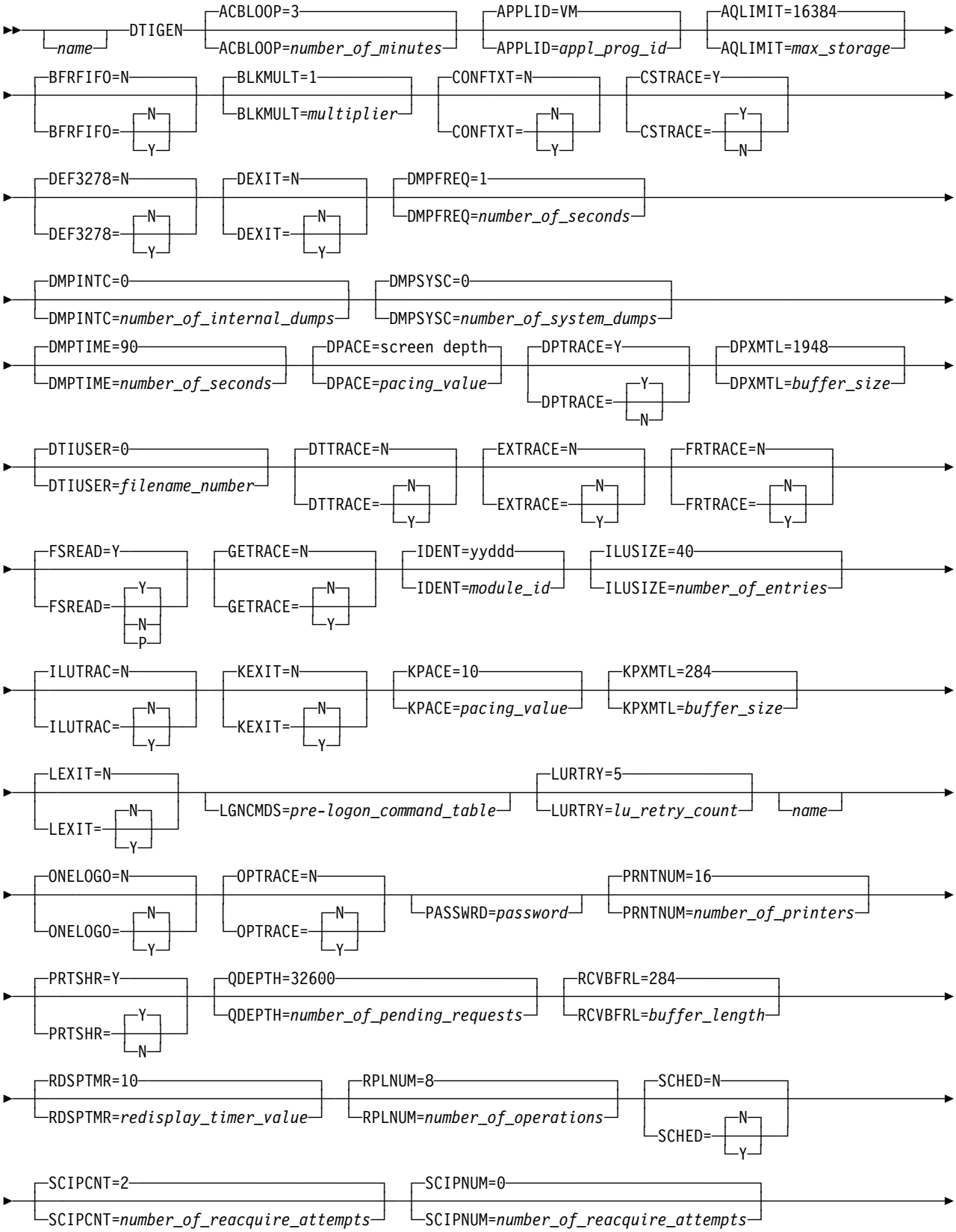
Installing VSCS Start Options

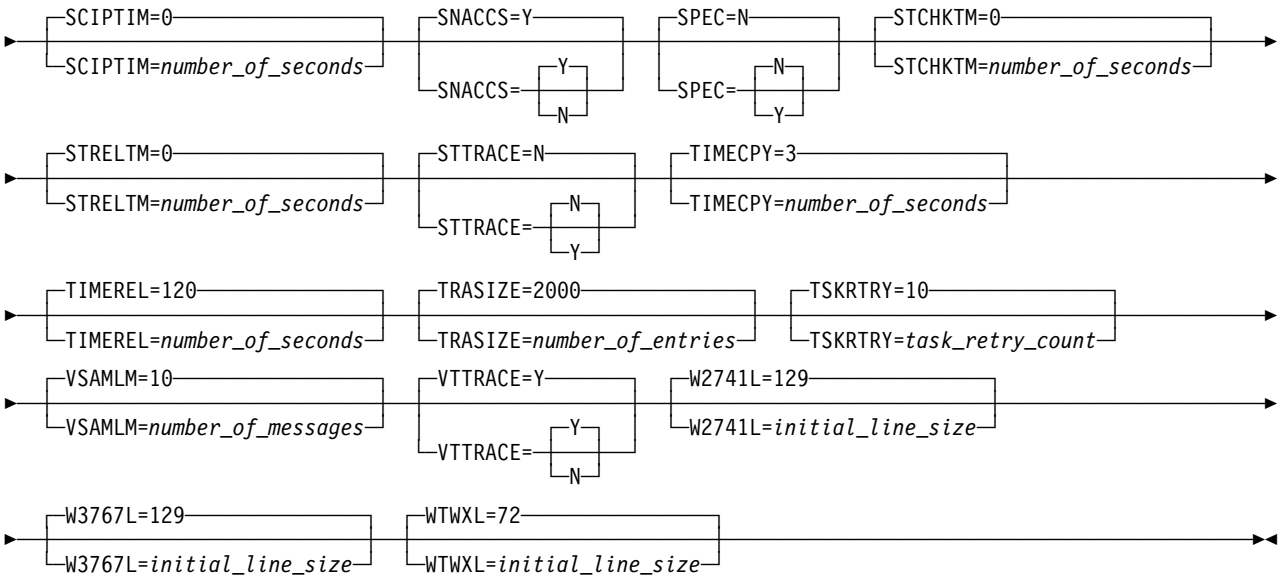
Refer to “Installing Tables and Modules in VM” in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit a set of VSCS start options.

Full Syntax

The full syntax for the VSCS application program follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

DTIGEN

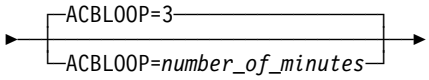




Operand Descriptions

The following section describes the operands you can code on the DTIGEN macroinstruction.

ACBLOOP



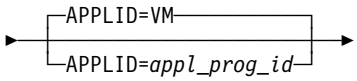
dependency: none
range: 0–32767

specifies the amount of time, in minutes, that VSCS retries opening its VTAM ACB after certain error conditions occur. VSCS retries every 30 seconds until one of the following is true:

- OPEN is successful.
- OPEN fails with a condition that cannot be retried.
- The retry limit is exceeded.
- VSCS is terminated by the VSCS operator command HALT, QUIT, or CANCEL.

Note that 0 specifies that VSCS will not retry the operation.

APPLID



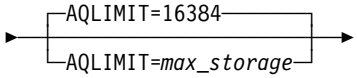
dependency: ACBNAME name

specifies the VTAM application program identifier. VSCS uses this operand for the VTAM OPEN ACB macroinstruction. It is also used as the IUCV application program identifier. *appl_prog_id* can be 1–8 alphanumeric characters. The first character must be alphabetical.

This value must match the name given for the ACBNAME operand of the VTAM APPL definition statement that defines VSCS. Refer to the *VTAM Network Implementation Guide* for more information on defining the VSCS application.

To tailor this name for LUs logging on to VSCS, use an interpret table or a USS table. See Chapter 6, “User-Defined Tables and Data Filter” on page 635 for details.

AQLIMIT



dependency: SCHED=Y
range: 8192–32768

specifies the maximum amount of storage for exception response mode PIUs (PIUs that do not require a definite response) that are queued for each LU. The larger the value, the more exception response PIUs VSCS generates, but also the more private storage VSCS uses.

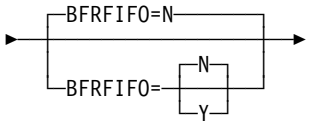
VSCS saves PIUs when operating in exception response mode so that if one or more fail it can recover by retransmitting the failing PIUs. Once this limit is reached, VSCS sends out the next PIU requesting a definite response. This allows it to release all of these queued PIUs upon receipt of the definite response.

Any value that is not a multiple of 1024 is rounded up to the next 1024 multiple. Specifying a value less than the default causes VSCS to use more definite response SENDs. This lessens the benefits of running in exception response mode (SCHED=Y) and consequently should only be done under situations where VSCS private storage is constrained.

This option is valid only when running with the SCHED=Y option (exception response mode).

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

BFRFIFO



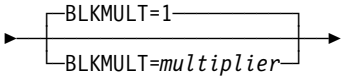
dependency: could affect performance

specifies whether dynamic buffers are selected from the dynamic buffer storage queues on a first-in first-out (FIFO) basis. The default is N, meaning that the buffers are selected on a last-in first-out (LIFO) basis.

See the description of the BLKMULT operand, for information on expanding the number of buffers.

Note: Code BFRFIFO=Y only when it is required for problem determination; if you want to see information in the last-used control block, for example, BFRFIFO=Y prevents VSCS from immediately reusing that area of storage. Use of this option could affect system performance. It should normally be initiated by an operator command when necessary. Refer to *VTAM Diagnosis* for information on using BFRFIFO=Y.

BLKMULT



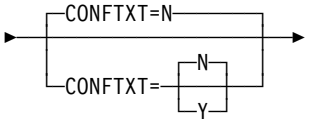
dependency: none
range: 1–8

specifies the multiplier to determine the size of the storage segments obtained from the system.

The recommended value is 1. While a larger value reduces the number of storage requests, it also reduces the ability of the VSCS storage manager to swap or release available storage segments. For example, if BLKMULT=3, storage pools that normally request 4K storage segments would request 12K storage segments.

Refer to the *VTAM Network Implementation Guide* for more information on controlling VSCS storage usage.

CONFTXT



dependency: none

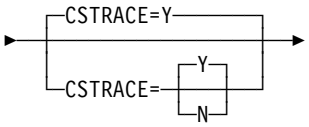
specifies whether tracing of user buffer data should be suppressed.

CONFTXT=N
specifies that both VTAM and VSCS can trace user buffer data.

CONFTXT=Y
prevents both VTAM and VSCS from tracing user buffer data.

You can change this operand while VSCS is active by using the VSCS CHANGE command. For VSCS user buffer data tracing, changes made with this command immediately affect all sessions. However, for VTAM tracing, changes made with this command affect only those sessions that are initiated after the command is issued. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

CSTRACE



dependency: could affect performance

specifies whether the console communications services (CCS) trace is on or off. The CCS trace traces the data flow between VSCS and CCS.

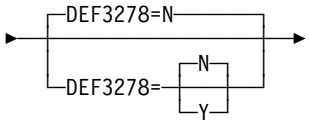
CSTRACE=N
turns off the CCS trace.

CSTRACE=Y
leaves the CCS trace on.

For more information on this trace, refer to *VTAM Diagnosis*.

Note: Use of this trace option could affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

DEF3278



dependency: none

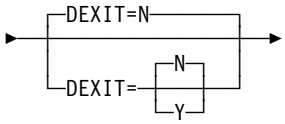
specifies whether the default device type for LU type 0 is 3277 or 3278 at logon time.

DEF3278=N
specifies that all LU type 0 devices are not assumed to be 3278.

DEF3278=Y
specifies that all LU type 0 devices are assumed to be 3278 unless the X'10' bit is on in the logmode used.

This takes the place of specifying X'40' in the logmode, which will allow logmodes from MVS to provide basic 3278 support on VM without alteration. This is similar to the way that VSCS treats LU type 2 and 3 devices.

DEXIT



dependency: none

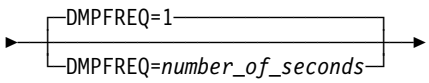
specifies whether to activate the VSCS data manipulation exit routines that translate data being sent to and from display devices. If you want to code these routines, refer to *VTAM Customization*.

DEXIT=N
specifies that the exit routines for display terminals are not activated.

DEXIT=Y
specifies that the exit routines for display terminals are activated.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DMPFREQ



dependency: none
range: value of DMPTIME–86399

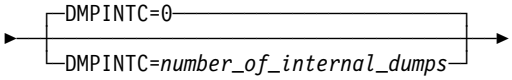
specifies the number of seconds in which only DMPINTC repetitive internal dumps and DMPSYSC repetitive system dumps are allowed. This operand can be used to control the number of dumps taken by VSCS.

Note that 0 specifies that the time interval is set to the entire time that VSCS is active. Thus, after the DMPINTC or DMPSYSC limit is reached for a specific error condition, no more dumps are taken for that error condition.

Refer to the *VTAM Network Implementation Guide* for information on using this operand to control the number of repetitive dumps taken by VSCS.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DMPINTC



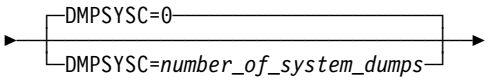
dependency: none
range: 0–255

specifies the number of repetitive internal dumps that are allowed during the DMPFREQ time interval. Internal dumps are dumps that are requested by VSCS when an error condition is detected. Internal dumps are repetitive if they are requested for the same internal error condition.

Note that 0 specifies that there is no limit to the number of repetitive internal dumps that are allowed.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DMPSYSC



dependency: none
range: 0–255

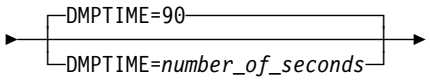
specifies the number of repetitive system dumps that are allowed during the DMPFREQ time interval. System dumps are dumps that are requested by the VSCS ESTAE routine, which receives control after a VSCS task ends abnormally. System dumps are repetitive if they have the same abend completion code. However, less common abend completion codes are grouped together so that dumps with these completion codes are considered repetitive.

Note that 0 specifies that there is no limit to the number of repetitive system dumps that are allowed.

Refer to the *VTAM Network Implementation Guide* for information on using this operand to control the number of repetitive dumps taken by VSCS.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DMPTIME

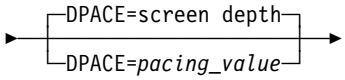


dependency: none
range: 1–86399

specifies the number of seconds required to take a dump of the VSCS virtual machine. This operand is used to validate the range of DMPFREQ, ensuring that the value of DMPFREQ is not less than the value of DMPTIME. If the time specified for DMPFREQ were less than the time required to dump the VSCS virtual machine, VSCS would allow all requested dumps to be taken, thus defeating the purpose of the DMPFREQ, DMPINTC, and DMPSYSC operands.

Refer to the *VTAM Network Implementation Guide* for information on controlling the number of repetitive dumps taken by VSCS.

DPACE



dependency: none
range: 0–255

specifies the pacing value between CCS and VSCS to be used with display terminals. The value of this operand determines how many messages CCS sends to VSCS before waiting for a response.

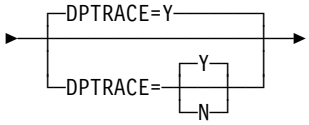
The default is the number of rows on the display terminal (the screen depth), and it is device-dependent. It is strongly recommended that you use the default.

If DPACE is not specified or set to 0, DPACE is set to the screen depth of the display during logon. If DPACE is set to a non-zero value, that value is used for all display devices without regard to screen depth.

Note: This pacing option is not related to the PACING and VPACING options available in VTAM and NCP.

You can change this operand while VSCS is active by using the VSCS CHANGE command. This command affects only those sessions that are initiated after the command is issued. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DPTRACE



dependency: could affect performance

specifies whether the VSCS dispatcher trace is on or off. This trace records information about VSCS before it dispatches control to the routines that process the various tasks.

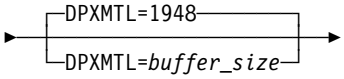
DPTRACE=N
specifies that the VSCS dispatcher trace is turned off.

DPTRACE=Y
specifies that the VSCS dispatcher trace is turned on.

For more information on this trace, refer to *VTAM Diagnosis*.

Note: Use of this trace option could affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

DPXMTL



dependency: none
range: 256–32600

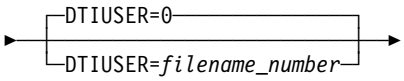
specifies the buffer size, in bytes, for data being sent to display terminals in line mode. (For terminals operating in full-screen mode, VSCS uses an internal mechanism to allocate buffers that are large enough to accommodate the full screen of data being sent.)

In general, 1948 should be the average size of the data stream transmitted to a display terminal operating in line mode. 1948 is the size of a 24 × 80 screen, minus the input area and the status area. This is usually the average number of lines of output sent to the terminal before a RECEIVE or READ is requested.

To determine the best value for your system, calculate the average buffer size by using the VSCS accounting records for a representative sample of users. Each VSCS accounting record contains the number of console output lines transmitted and the number of output requests; use these values to calculate the average number of console lines for each VTAM SEND request. Refer to the *VTAM Network Implementation Guide* for the format of the VSCS accounting record.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

DTIUSER

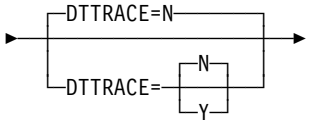


dependency: none
range: 0–9

specifies the name of the ASSEMBLE file that contains this DTIGEN macroinstruction. The DTIUSER value should appear in the filename (that is, DTIUSER*n* ASSEMBLE).

This operand is used only for diagnostic purposes, to determine which set of initialization parameters and VSCS options were used.

DTTRACE



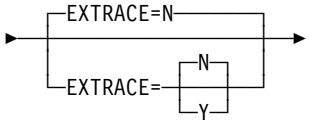
dependency: none

specifies whether the VSCS data trace is on or off. This option traces user data transmitted to and received from a user's LU. Data is traced using the format received from CCS (outbound) or in the format provided to CCS (inbound). Internal trace will capture 20 bytes of data; external trace will capture up to 243 bytes of data.

DTTRACE=N
specifies that the VSCS data trace is turned off.

DTTRACE=Y
specifies that the VSCS data trace is turned on.

EXTRACE



dependency: could affect performance

specifies whether external tracing is used.

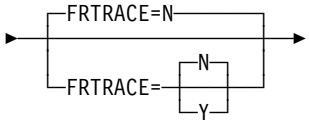
EXTRACE=N
specifies internal tracing only.

EXTRACE=Y
specifies that external tracing is used if the system external trace facility has been activated.

For more information on external and internal tracing, refer to *VTAM Diagnosis*.

Note: Use of this trace option could affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

FRTRACE



dependency: could affect performance

specifies whether the VSCS free block trace is used. This option traces requests to the VSCS storage manager to release a block of storage. This trace entry is not written to individual LU trace tables. It is only written externally (if EXTRACE=Y has been specified) and to the VSCS global trace table.

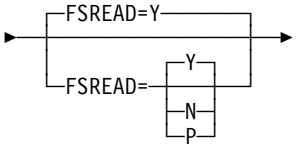
FRTRACE=N
specifies that the VSCS free block trace is not used.

FRTRACE=Y
specifies that the VSCS free block trace is used.

For more information on this trace, refer to *VTAM Diagnosis*.

Note: Use of this trace option might affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

FSREAD



dependency: applies only to display terminals

specifies whether VSCS can issue an internal full-screen READ when a mode switch is required.

FSREAD=Y
specifies that VSCS can issue a full-screen 3270 READ buffer command when a switch from full-screen mode is required. A terminal in full-screen mode requires a mode switch to process a control program (CP)-generated console mode write request (for example, a CP MSG command) or a CP READ. Specifying FSREAD=Y can prevent the loss of input data for application programs by retrieving READ buffer data from a logical unit.

FSREAD=N
specifies that VSCS does not issue an internal full-screen READ when a mode switch is required. This could cause data to be lost for applications that retrieve input data with a READ buffer.

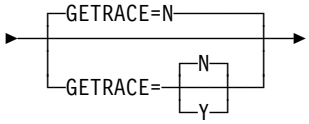
FSREAD=P
specifies that VSCS can issue a full-screen 3270 READ buffer command, but the screen-saving function is only initiated when VSCS receives input generated when a user presses a PAKEY (PA1, PA2, PA3).

Coding FSREAD=Y or FSREAD=P can prevent the loss of input data for application programs by retrieving READ buffer data from the terminal. However, since FSREAD=P only retrieves the screen data after receiving PAKEY input, applications that issue READ buffer commands after receiving other than PAKEY input (for example, PFKEY, ENTER key) might find the READ buffer commands failing due to a switch to console mode.

Note: The 3270 READ buffer command causes a large amount of data to be transmitted. This might result in an undesirable delay for the user. If the application does request the READ buffer data from VSCS, then this delay would have occurred anyway. However, if the application does not request the READ buffer data from VSCS, then the additional delay caused by VSCS issuing the 3270 READ buffer command might be unacceptable.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

GETRACE



dependency: could affect performance

specifies whether the get block trace is used. This option traces requests to the VSCS storage manager for a block of storage. This trace entry is not written to individual LU trace tables. It is only written externally (if EXTRACE=Y has been specified) and to the VSCS global trace table.

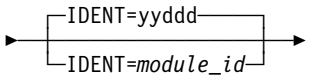
GETRACE=N
specifies that the VSCS get block trace is not used.

GETRACE=Y
specifies that the VSCS get block trace is used.

For more information on this trace, refer to *VTAM Diagnosis*.

Note: Use of this trace option might affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

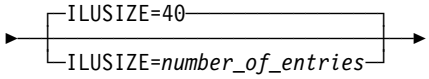
IDENT



dependency: none

provides 1–5 EBCDIC characters used to identify the DTIUSERx module that was the source of the original set of VSCS start options. Used only for debugging, this identifier appears in a dump of the VSCS virtual machine. The default is the Julian date (*yyddd*) on which the DTIGEN macroinstruction was assembled.

ILUSIZE



dependency: none
range: 5–125

specifies the number of 32-byte entries in each LU's individual trace table. Since VSCS allocates storage in a number of predefined block sizes, the actual size of each LU's trace table might be somewhat larger. A VSCS DISPLAY of an LU will show its actual trace table location and size. Each individual LU trace table will be positioned on a 16-byte boundary for readability in a dump.

If you specify a size:

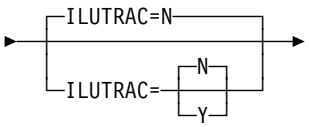
Greater than or equal to	and less than or equal to	you will get a maximum value of	storage used (bytes)[1]
5	6	7	256
7	9	9	360

Greater than or equal to	and less than or equal to	you will get a maximum value of	storage used (bytes)[1]
10	16	16	568
17	23	24	800
24	40	40	1344
41	49	49	1616
50	61	62	2024
62	74	75	2440
75	93	94	3048
94	125	126	4064

Note:

1. Storage used is per LU.

ILUTRAC



dependency: none

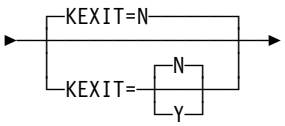
specifies whether trace entries specific to an LU are written to an LU's individual trace table. Each LU, when it logs onto VSCS, will have a trace table allocated that is at least as large as specified in the ILUSIZE operand. The VSCS global trace table is still maintained as before, however entries specific to each LU are also written to each LU's own individual trace table.

ILUTRAC=N
specifies that trace entries are not written to an LU's trace table.

ILUTRAC=Y
specifies that trace entries are written to an LU's trace table.

For more information on this tracing mode, refer to *VTAM Diagnosis*.

KEXIT



dependency: none

specifies whether VTAM is to activate the VSCS data manipulation exit routines that translate data being sent to and from keyboards/printers and TWX devices.

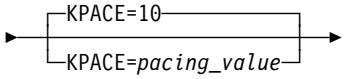
KEXIT=N
specifies that the exit routines for these devices are not active.

KEXIT=Y
specifies that the exit routines for these devices are active.

If you want to code these routines, refer to *VTAM Customization*.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

KPACE



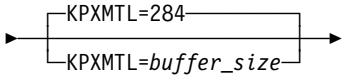
dependency: none
range: 1–255

specifies the pacing value between CCS and VSCS to be used with keyboards or printers. The value specifies how many console mode messages CCS sends to VSCS before waiting for a response. Note that CP might not respond to any of the attention keys until this number of messages has been sent to the device. It is strongly recommended that you use the default.

You can change this operand while VSCS is active by using the VSCS CHANGE command. This command affects only those sessions that are initiated after the command is issued. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

Note: This pacing option is not related to the PACING and VPACING options available in VTAM and NCP.

KPXMTL



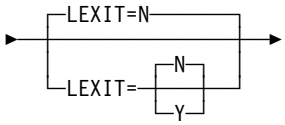
dependency: none
range: 138–32600

specifies the buffer size, in bytes, for data being sent to keyboards/printers. In general, this value should be the average size of the data stream transmitted to the device.

Note: For TWX-compatible display terminals (such as an IBM Personal Computer emulating a 3101), you should determine whether any application programs send large blocks of data. If so, you might need a larger value for KPXMTL.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

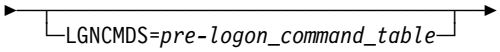
LEXIT



dependency: none

specifies whether the VSCS logon exit (DTIPRLOG) will be given control to process all logon requests.

LGNCMDS



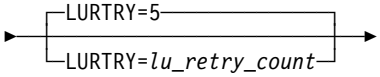
dependency: none

specifies the name of a table that contains a list of the commands a user can issue before logging on. When VSCS receives data from a terminal that is not logged on, it compares the data with the commands in this table. If a match occurs, VSCS passes the command to CP. If no match occurs, VSCS assumes the data is a user ID (followed by options, if any) and appends it to the first command in the table before passing it to CP. Therefore, the first command in the table must be LOGON or some other command that allows access to the system.

You must code the basic list of commands shown in the following table, plus any others you wish to add. The largest number of commands you can have is 64. The basic command table is:

```
(LOGON,L,LOGIN,SL,SLEEP,LOG,LOGOFF,M,MSG,MESSAGE,D,DIAL,*)
```

LURTRY



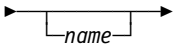
dependency: none
range: 1–the value of TSKRTRY

is the logical unit retry count. If a specific logical unit causes repeated error conditions (abends, internal errors or VTAM errors), this count tells how many times reactivation should be attempted. When the count is exceeded, VSCS disconnects the logical unit from VM and then releases it back to the SNA network.

The default is 5 or the TSKRTRY value, whichever is smaller. Although the value of TSKRTRY is the upper limit, this number should be less than the TSKRTRY count so that one user cannot cause VSCS to terminate.

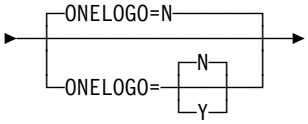
You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

name



name is optional. If you specify a name, it must be 1–8 alphanumeric characters, beginning with a character other than a dollar sign ($).

ONELOGO



dependency: none

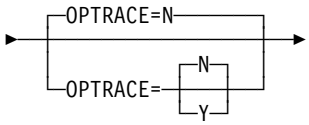
determines whether the VM logo is redisplayed after a terminal user logs off or disconnects from CP. This operand affects only terminal users that logged on to VM from the VM logo.

ONELOGO=N
specifies that the VM logo is redisplayed for terminal users who log off or disconnect from CP, if they originally logged on to VM from the logo.

ONELOGO=Y
specifies that the VM logo is not redisplayed for terminal users who log off or disconnect from CP, if they originally logged on to VM from the logo.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

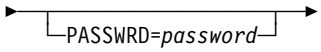
OPTRACE



dependency: none

specifies whether the trace OPER should be invoked. The trace indicates, in the VSCS trace table (OPER) or externally, when a VSCS command or a message is issued.

PASSWRD

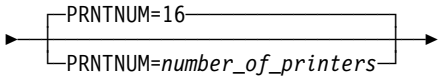


dependency: none

identifies the VTAM application password in 1–8 EBCDIC characters. If you code a PRTCT operand on the APPL definition statement for VSCS, this password must match the value of that PRTCT operand. (Refer to the *VTAM Network Implementation Guide* for information on coding the APPL definition statement for a VSCS application.). If you omit PRTCT on the APPL definition statement, VTAM does not check passwords.

The default is the value you coded for the APPLID operand on the DTIGEN macroinstruction.

PRNTNUM



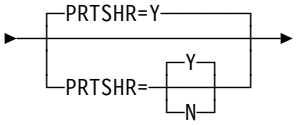
dependency: none
range: 1–32600

is the maximum number of printers you can define for VSCS PF-key copy requests. PRNTNUM determines the number of entries in the VSCS printer table.

Because each entry requires 40 bytes of storage, the value you choose should accommodate a normal increase in the number of printers, without wasting storage.

Refer to the *VTAM Network Implementation Guide* for information on how VSCS uses printers.

PRTSHR



dependency: none

specifies whether VSCS can dynamically reallocate and share printers.

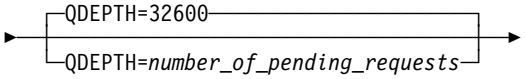
PRTSHR=N
specifies that VSCS does not release the printer when it is requested by another application program. In that case, the printer must be released by issuing the VTAM VARY INACT or VSCS FORCE command. Refer to the *VTAM Network Implementation Guide* for information on how VSCS can share printers with other applications.

PRTSHR=Y
specifies that VSCS shares printers with other application programs. Thus, when the VSCS RELREQ exit is driven (to indicate a request for a printer):

- If the time designated by the TIMEREL operand has elapsed since the printer's last use, VSCS immediately releases the printer.
- Otherwise, VSCS waits until the printer has been idle for the time specified by TIMEREL before releasing the printer.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

QDEPTH

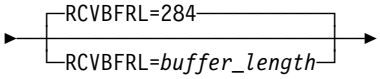


dependency: none
range: 200–32600

specifies the number of pending requests allowed for an LU. When the value of QDEPTH is exceeded, the LU is purged. VSCS monitors the number of events pending for each LU to help prevent storage shortages while LUs are in holding mode.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

RCVBFRL



dependency: none
range: 85–32600

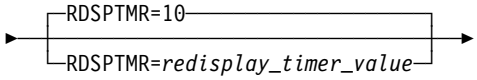
specifies the buffer length, in bytes, for VTAM RECEIVE requests issued by VSCS.

In general, RCVBFRL should be set to the average size of the data stream that VSCS receives from a VTAM logical unit. If the RECEIVE data exceeds this buffer size, VSCS dynamically allocates additional buffer space to receive the remaining data from VTAM.

To find the best value for your system, calculate the average buffer size by using the VSCS accounting records for a representative sample of users. Each VSCS accounting record contains the number of bytes received and the number of input requests; use these values to calculate the average buffer size. See the *VTAM Network Implementation Guide* for the format of the VSCS accounting record.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

RDSPTMR

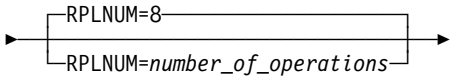


dependency: none
range: 0–255

specifies the redisplay timer value, in tenths of a second. RDSPTMR sets the time that elapses before input is redisplayed on the screen. This reduces the number of writes to the terminal. If output comes back during this time, it is displayed immediately; otherwise, the input is redisplayed, and the output is displayed when it arrives. The default is 10 (1 second), which is the recommended value. VSCS Timer Services uses a variable clock that allows timers to expire on time and not wait for a three second interval. If you set the value higher than 30 (3 seconds), users might think their terminals are hung, or complain of slow response time when they do complex operations.

Note: Although you code this value in tenths of a second, the assembling process translates it to sixteenths of a second. It is displayed in sixteenths of a second in message DTIV31I.

RPLNUM



dependency: none
range: 1–255

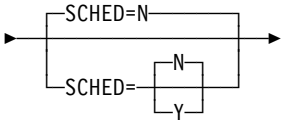
specifies the number of VTAM RECEIVE operations that VSCS continually has outstanding. The storage used by VSCS for such operations, on a continuous basis, is the RCVBFRL size times the value of RPLNUM. (VSCS might also be using additional buffer storage obtained by an internal mechanism for RECEIVE operations that exceed the RCVBFRL value and for full-screen output operations.)

The default, 8, is the recommended value. However, if you are running more than 40 terminals, you can increase the value to 255 RPLs.

Note: The RPLNUM affects only RECEIVE ANY RPLs. When switching to RECEIVE SPECIFIC (SPEC=Y), the RECEIVE ANY RPLs are only used once.

Each LU that is in session with VSCS will have its own RPL allocated, but not from the RECEIVE ANY RPL pool.

SCHED



dependency: none

specifies whether exception response VTAM SENDs should be used when sending output requests to a logical unit. This option affects all devices except printers.

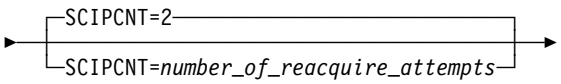
SCHED=N
specifies that definite response VTAM SENDs are used to send output to display logical units and keyboard/printer devices.

SCHED=Y
specifies that exception response VTAM SENDs are used to send output to display logical units and keyboard/printer devices, except for requests for synchronization, protocol requirements, and error recovery.

For SCHED=Y, additional storage is required for a limited error recovery process. This additional storage consists of saved exception response PIUs. The amount of this additional storage is controlled by AQLIMIT and the VSCS CHANGE command option and defaults to a maximum of 16K per exception response LU.

VSCS frees saved exception response PIUs that are unlikely to be needed (that is, PIUs that are older than 90 seconds) if a value greater than 0 is specified for the DTIGEN start option STCHKTM. Consequently, it is recommended that if SCHED=Y is specified, then a value greater than 0 be specified for STCHKTM.

SCIPCNT



dependency: none
range: 1–10

specifies the number of times an attempt should be made to reacquire a logical unit after a network failure.

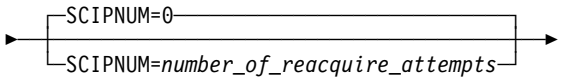
You should specify a value determined by your network setup. If there is more than one route to logical unit resources, a small number of attempts should be adequate. If multiple routes do not exist, a larger number of allowed attempts gives a greater likelihood of the logical unit's being reacquired after the network recovers.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

Notes:

1. If the network fails, a large number of recovery attempts with a large number of logical units can adversely affect VSCS performance. The operands SCIPNUM and SCIPTIM can be used to restrict the number of recovery attempts made by all logical units during a specified time interval. Refer to the *VTAM Network Implementation Guide* for information on controlling VSCS SCIP recovery.
2. For VSCS SCIP recovery to properly recover devices, the appropriate logon mode table and entry must be coded for each logical unit. Otherwise, VTAM uses the default logon mode entry, which might not be valid for the recovering logical unit.

SCIPNUM



dependency: none
range: 0–32600

specifies the number of attempts made during the time interval specified by SCIPTIM to reacquire the logical units that are in recovery. This operand can be used, with the operand SCIPTIM, to prevent the excessive overhead of recovery that is experienced when major network outages occur.

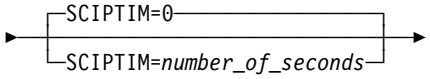
Multiple recovery attempts can be made for the same logical unit, each of which is counted to be checked for the SCIPNUM limit. The maximum number of recovery attempts made by a single logical unit is specified by the SCIPCNT operand.

If SCIPNUM=0 is coded or used by default, there is no limit to the number of recovery attempts.

See the *VTAM Network Implementation Guide* for more information and examples of using SCIPNUM to control VSCS recovery.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

SCIPTIM



dependency: none
range: 0–3600 (1 hour)

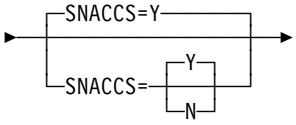
specifies the time interval, in seconds, in which no more than SCIPNUM attempts are made to reacquire the logical units that are in recovery. This operand can be used, with the operand SCIPNUM, to prevent the excessive overhead of recovery that is experienced when major network outages occur.

If SCIPTIM=0, SCIPCNT is treated as if it were 0, and there is no limit to the number of recovery attempts.

See the *VTAM Network Implementation Guide* for more information and examples of using SCIPTIM to control VSCS recovery.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

SNACCS

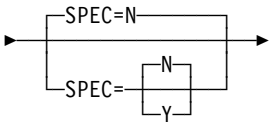


dependency: none

specifies whether VSCS automatically issues the ENABLE SNA command for the operator during VSCS initialization. Coding SNACCS=Y (or allowing the default) prevents VSCS from hanging because SNA is not enabled.

Note that SNACCS cannot be modified with the VSCS CHANGE command.

SPEC



dependency: none

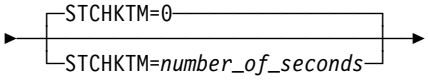
specifies the mode used for VTAM RECEIVE processing of input from logical units. This operand applies to all types of devices, including printers.

SPEC=N
specifies that VTAM RECEIVE ANYs are used to receive input from any logical unit. However, at times, a VTAM RECEIVE SPECIFIC is required to receive input from a specific logical unit. The number of RECEIVE ANYs used by VSCS is controlled by the RPLNUM operand.

SPEC=Y
specifies that a VTAM RECEIVE SPECIFIC is used for each logical unit so that input can be received from that specific logical unit. This requires additional storage, because each logical unit requires a RECEIVE RPL and receive buffer. The size of the receive buffer is determined by the value specified on the RCVBFRL operand.

Note: The RPLNUM affects only RECEIVE ANY RPLs. When switching to RECEIVE SPECIFIC (SPEC=Y), the RECEIVE ANY RPLs are only used once. Each LU that is in session with VSCS will have its own RPL allocated, but not from the RECEIVE ANY RPL pool.

STCHKTM



dependency: none
range: 0–86399

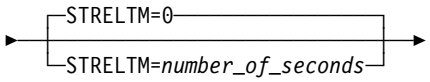
specifies the number of seconds between storage checks. A storage check involves scanning the VSCS storage pools for segments in which all blocks are free (available segments), and removing half of those segments from each storage pool. The available segments that are removed from the storage pools are made

accessible to satisfy requests for all storage pools and also made accessible for processing during the next storage release. See the description of the STRELTM operand for information about storage releases. Refer to the *VTAM Network Implementation Guide* for a discussion of how to control VSCS storage management.

0 means that no storage checking is done.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

STRELTM



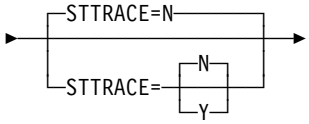
dependency: none
range: 0–86399

specifies the number of seconds between storage releases. A storage release involves freeing half of the remaining available storage segments that were removed from the storage pools during the last storage check. See the description of the STCHKTM operand for information about storage checks. Refer to the *VTAM Network Implementation Guide* for a discussion of how to control VSCS storage management.

0 means no storage releasing is done.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

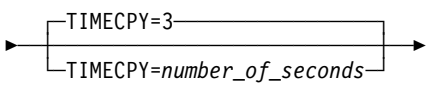
STTRACE



dependency: none

specifies whether the trace STAT should be invoked. This trace provides a method for monitoring VSCS storage use over a period. STTRACE only records externally and must be externally started. Changes to most of the DTIGEN start options are recorded.

TIMECPY

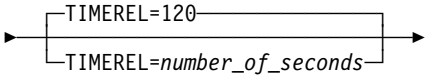


dependency: none
range: 0–30

defines the printer allocation time limit in seconds. This is the maximum time that VSCS waits to acquire a printer before causing a PF-key copy request to fail.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

TIMEREL



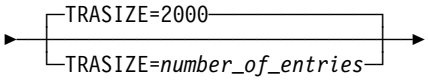
dependency: PRTSHR=Y
range: 0–86399

defines the printer deallocation time limit in seconds. If a printer has been requested by another application, VSCS releases the printer only after it has been idle for the number of seconds specified by TIMEREL.

This operand is ignored if PRTSHR=N.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

TRASIZE



dependency: none
range: 125–64000

specifies the number of 32-byte entries in the trace table. This number should be large enough to record data from all the VSCS traces (CSTRACE, DPTRACE, EXTRACE, FRTRACE, and VTTRACE) that are active concurrently.

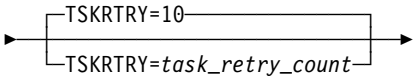
The trace table is positioned on a 32-byte boundary for readability in a storage dump. The total size of the trace table in bytes is:

$$(\text{TRASIZE} + 3) \times 32$$

This storage is allocated when the first trace entry is made and is freed only when VSCS is terminated.

Note: When issuing the VSCS TRACEON command, the operator can request that trace data be recorded on an external medium.

TSKRTRY



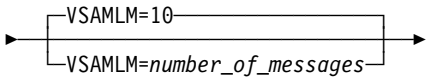
dependency: value must be greater than LURTRY value
range: 1–255

specifies the task retry count (maximum number of retries of a task that abends) before VSCS termination is invoked.

The value of TSKRTRY must be greater than the LURTRY count.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

VSAMLM

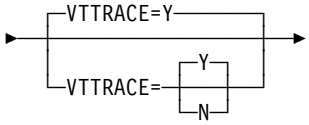


dependency: none
range: 4–255

specifies the maximum number of concurrent messages for an LU that can be transferred between VSCS and CCS using IUCV. For each LU, VSCS initiates communication with CCS by using IUCV CONNECT.

You can change this operand while VSCS is active by using the VSCS CHANGE command. This command affects only those sessions that are initiated after the command is issued. Refer to *VTAM Operation* for more information on the VSCS CHANGE command.

VTTRACE



dependency: could affect performance

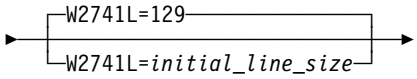
specifies whether the VSCS VTAM exit trace is used. This trace traces incoming requests and responses from VTAM at the application program interface (API). It also includes dispatcher trace entries for VTAM services.

VTTRACE=N
specifies that the trace is not used.

VTTRACE=Y
specifies that the trace is used. Y is the recommended value.

Note: Use of this trace option could affect performance. It should normally be initiated by the operator command VSCS TRACEON (option) when necessary.

W2741L



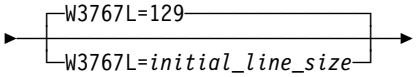
dependency: none
range: 1–255

specifies the initial line size, in characters, for 2741 terminals. This initial line size is used by VSCS during logon processing. Once the logon processing is complete, the CP TERMINAL setting determines the line size for the terminal.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command. This command affects only those terminal users who log on to CP (not those who reconnect) after the command is issued.

A terminal user logged on to VSCS can issue the CP TERMINAL LINESIZE command to change the value of the current line size for that device.

W3767L



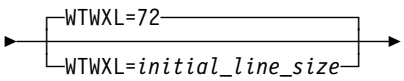
dependency: none
range: 1–255

specifies the initial line size, in characters, for 3767 or 3777 terminals. This initial line size is used by VSCS during logon processing. After the logon processing is complete, the CP TERMINAL setting determines the line size for the terminal.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command. This command affects only those terminal users who log on to CP (not those who reconnect) after the command is issued.

A terminal user logged on to VSCS can issue the CP TERMINAL command to change the value of the current line size for that device.

WTWXL



dependency: none
range: 1–255

specifies the initial line size, in characters, for TWX keyboards/printers. This initial line size is used by VSCS during logon processing. After the logon processing is complete, the CP TERMINAL setting determines the line size for the terminal.

You can change this operand while VSCS is active by using the VSCS CHANGE command. Refer to *VTAM Operation* for more information on the VSCS CHANGE command. This command affects only those terminal users who log on to CP (not those who reconnect) after the command is issued.

A terminal user logged on to VSCS can issue the CP TERMINAL command to change the value of the current line size for that device.

Coding Logon Mode Tables for VSCS Devices

Logon mode tables are necessary to define session parameters for devices supported by VSCS. To provide support for devices not supported by the IBM-supplied logon mode tables for VM and VSCS, you must customize these tables. In particular, you might have to code the MODEENT macroinstruction for SNA and non-SNA 3270 devices (such as the IBM 3767 and 3770). You might also need additional information to code the PSERVIC operand of this macroinstruction.

Use the MODETAB and MODEENT macroinstructions to define device characteristics in logon mode tables and their entries. The following are examples of definitions using MODEENT table entries.

MODEENT Macroinstruction for Non-SNA 3270 Devices

Following is an example of the MODEENT macroinstruction coding for non-SNA devices. Note that the FMPROF, TSPROF, PRIPROT, SECPROT, and COMPROT values shown are the same as those used in the IBM-supplied logon mode table, ISTINCLM, in the logon mode entry IBMS3270.

```
name            MODEENT FMPROF=X'02',                                  C
                      TSPROF=X'02',                                    C
                      PRIPROT=X'71',                                   C
                      SECPROT=X'40',                                   C
                      COMPROT=X'2000',                                 C
                      PSERVIC=x'00......................'
                          (see Table 70 on page 632)
```

MODEENT Macroinstruction for SNA LU Type 1 Devices

LU type 1 terminals include the IBM 3767 and 3770. When using the Network Terminal Option, the IBM 2741, IBM 3101, and TWX Models 33 and 35 are also supported as LU type 1 devices. Note that the distinction between LU type 1 and LU type 2 devices is indicated in the PSERVIC field of the logon mode table entry. The values used are the same as those in the IBM-supplied logon mode entries VSCS3767, VSCS2741, and VSCSTWX.

```
name            MODEENT FMPROF=X'03',                                  C
                      TSPROF=X'03',                                    C
                      PRIPROT=X'B1',                                   C
                      SECPROT=X'90',                                   C
                      COMPROT=X'3080',                                 C
                      PSERVIC=X'01......................'
                          (see Table 70 on page 632)
```

MODEENT Macroinstruction for SNA 3270 Devices

The logon mode entry values used in the following example are the same for both LU type 2 devices and LU type 3 printers with the exception of the LU type specification in the PSERVIC field. The RUSIZES operand value of X'8587' defines a 256-byte maximum secondary logical unit send size and a 1024-byte maximum primary logical unit send size. The RUSIZES value depends on the control unit and printer buffer specifications.

```
name            MODEENT FMPROF=X'03',                                  C
                      TSPROF=X'03',                                    C
                      PRIPROT=X'81',                                   C
                      SECPROT=X'90',                                   C
                      COMPROT=X'3080',                                 C
                      RUSIZES=X'8587',                                 C
                      PSERVIC=X'......................'
                          (see Table 70 on page 632)
```

PSERVIC Operand of the MODEENT Macroinstruction

The PSERVIC operand of the MODEENT macroinstruction defines the device LU type, buffer sizes, and QUERY capability (programmed symbols, extended color, or extended highlight support).

Code the 12 bytes of device-specific hexadecimal data of the PSERVIC operand as follows:

PSERVIC Coding	Description
X'00..00000000.........00'	For non-SNA (LU type 0)
X'01..00000000.........00'	For SNA LU type 1
X'02..00000000.........00'	For SNA LU type 2
X'03..00000000.........00'	For SNA LU type 3
X'..00.................'	Device with old APL and without extended data stream capability
X'..20.................'	Device is a VSCS printer.
X'..40.................'	Device with new APL
X'..80.................'	Device with extended data stream capability
X'..C0.................'	Device with new APL and with extended data stream capability
X'............0C5000007E..'	Buffer size 960 only (12X80)
X'............185020507F..'	Buffer size 1920 or 2650 (24X80 or 32X80)
X'............18502B507F..'	Buffer size 1920 or 3440 (24X80 or 43X80)
X'............18501B847F..'	Buffer size 1920 or 3564 (24X80 or 27X132)
X'............18503EA07F..'	Buffer size 1920 or 9920 (24X80 or 62X160)
X'..................00..'	Undefined row and column format
X'..................01..'	12 rows, 40 columns format
X'..................02..'	24 rows, 80 columns format
X'..................03..'	Get alternate screen size from WSFQ reply.
X'..................7E..'	Presentation space has a fixed primary (bytes 6, 7) size as defined in the PSERVIC field.
X'..................7F..'	Presentation space has both primary (bytes 6, 7) and alternate (bytes 8, 9) sizes as defined in the PSERVIC field.

The format of the PSERVIC field is shown in Table 70.

Table 70 (Page 1 of 2). Format of the PSERVIC Field

Byte	Description
0	LU type 0, 1, 2, or 3

Table 70 (Page 2 of 2). Format of the PSERVIC Field

Byte	Description
1	Used by VSCS **Value** **Meaning** X'00' Device with old APL and without extended data stream capability X'80' Device has extended data stream capability. It supports write structured field (WSF) and read partition (Query) structured field. X'40' Device with new APL[1] (not valid for LU types 2 and 3) **Note:** VSCS uses this bit to define the display LU type to VM: ON 3278 or 3279 (most are new APL)[1] OFF 3277 **Note:** If DEF3278=Y is coded in DTIGEN, then the X'40' bit is assumed and does not need to be coded. To logon a 3277 with DEF3278, you must turn on the X'10' bit, which will indicate the device is really a 3277, not 3278 which DEF3278=Y makes the default. X'20' Device is a VSCS printer. X'10' LU type 0 on an SNA control unit, used for a 3277 with graphics attachment only **Note:** This bit will indicate a 3277 type device if DEF3278=Y and device is an LU type 0. X'08' Write Structured Field (WSF) 3270 data stream (3270DS) command is not supported. X'88' LU type 0 or type 2 X'C8' LU type 0 with new APL only X'C0' Device with extended data stream capability and new APL[1] (not valid for LU types 2 and 3)
2–5	Zero
6–7	Primary size or zero
8–9	Alternate size or zero
10	Presentation space size (X'00', X'01', X'02', X'03'[2] , X'7E', X'7F')
11	Zero

Notes:

1. The APL type for extended data stream devices is set by the graphic escape character (X'08') support indicated in the Character Sets Query Reply. The APL type will not affect the device type.
2. Presentation space size X'03' is used for 3290 terminals with a screen larger than the standard size. If the value of X'03' is specified, a default primary screen size of 24x80 is established and VSCS issues a Write Structured Field Query (WSFQ) to the terminal to determine the alternate screen size that is used. If the value of X'03' is not specified, the primary and alternate screen sizes must be specified.

Chapter 6. User-Defined Tables and Data Filter

About This Chapter

This chapter describes how to create or modify the following tables and filter:

- Associated LU table
- Subarea and APPN class-of-service (COS)
- Interpret table
- Logon mode table
- Message-flooding prevention table
- Model name table
- Session awareness (SAW) data filter
- USS table.

Follow the assembler language conventions described in “Format of Macroinstructions” on page 11 when coding the macroinstructions discussed in this chapter.

The MODIFY TABLE operator command lets you dynamically replace COS, interpret, logon mode, USS, associated LU, and model name tables and the session awareness (SAW) data filter. Refer to“MODIFY TABLE Command” in *VTAM Operation* for information on the MODIFY TABLE command.

MVS To assemble the tables, your JCL will need to point to the SYS1.SISTMAC1 macroinstruction library.

VM Refer to “Installing Tables and Modules in VM” in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit the tables.

VSE To assemble the tables, your JCL will need to point to the VTAM sublibrary.

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 683.

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 Chapter 2, “Major Nodes.”

Table 71. Associated LU Table Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	ASLTAB					
name	ASLENT	PRINTER1 PRINTER2	640 640			
name	ASLPLU	PLU PRINTER1 PRINTER2	639 640 640	R		

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

MVS Store the set of macroinstructions as a member of SYS1.VTAMLST and use the member name as the name of the table.

VSE Store the set of macroinstructions in the private definition library referenced in the VTAM start procedure.

VM Store the macroinstructions with a filetype of VTAMLST and use the filename as the name of the table.

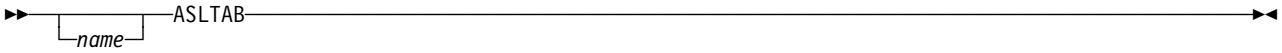
No assembly or link-edit is required to install this table. The table will be dynamically built upon the first activation of any resource that has a defined association to the table. When the last resource using it is deactivated, the table is deleted and its storage is freed.

IBM does not supply a default associated LU table.

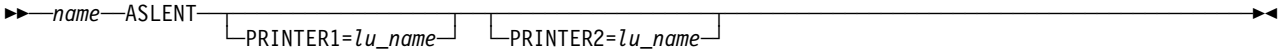
Full Syntax

The full syntax for the associated LU table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

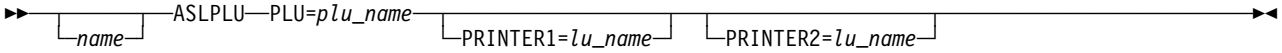
ASLTAB



ASLENT



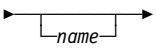
ASLPLU



Operand Descriptions

The following section describes the operands you can code for the associated LU table.

name



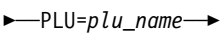
macroinstructions: ASLTAB, ASLPLU

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

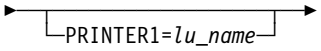


macroinstructions: ASLPLU
dependency: none

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.

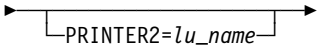
PRINTER1



macroinstructions: ASLENT, ASLPLU
dependency: none

specifies the 1–8 character name of the primary printer associated with the terminal as identified in the SLU's network.

PRINTER2



macroinstructions: ASLENT, ASLPLU
dependency: 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 72. Subarea COS Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	COSTAB					
name	COS	VR SUBSTUT	643 643	R	 NO	see description
	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.

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 (MVS, VM), 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.

MVS This table should be link-edited as a nonexecutable module into a library associated with the SYS1.VTAMLIB DD statement.

VM Refer to “Installing Tables and Modules in VM” in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit this table.

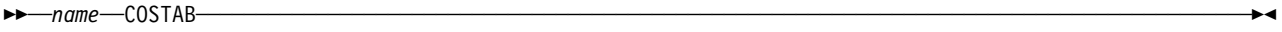
VSE Link-edit the module into a private definition library referenced in the VTAM start procedure.

VTAM loads the resulting load module during initialization and uses it later to establish sessions.

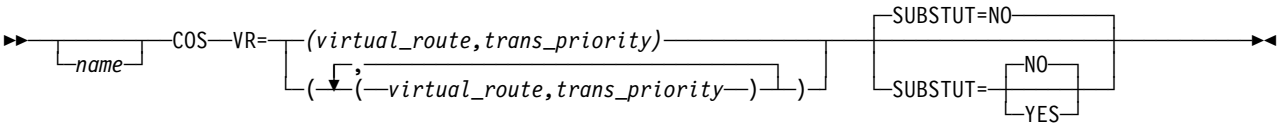
Full Syntax

The full syntax for the subarea class of service table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

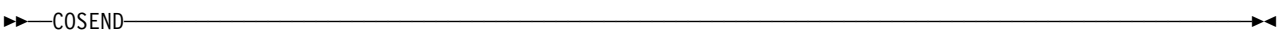
COSTAB



COS



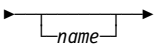
COSEND



Operand Descriptions

The following section describes the operands you can code for the subarea class of service table.

name



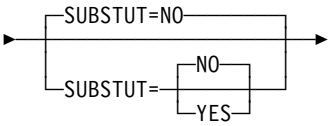
macroinstructions: COSTAB

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



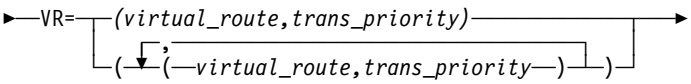
macroinstructions: COS
dependency: none

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 “Session Management Exit Routine” in *VTAM Customization* for more information on the session management exit routine, and to “How Session Traffic Is Assigned to a Specific Route” in the *VTAM Network Implementation Guide* for more information on substituting class-of-service values.

VR



macroinstructions: COS
dependency: 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"Virtual Route Selection Exit Routine" in *VTAM 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 73. APPN COS Definition Statements and Operands

Name	Definition Statement	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	APPNCOS	PRIORITY	649	R		MEDIUM	
	LINEROW	NUMBER	648	R			
		CAPACITY	645				
		COSTBYTE	646				
		COSTTIME	647				
		PDELAY	649				
		SECURITY	650				MVS,VM
		UPARM1	651				
		UPARM2	651				
		UPARM3	651				
		WEIGHT	652				
	NODEROW	NUMBER	648	R			
		CONGEST	646				
		ROUTERES	650				
		WEIGHT	652				

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

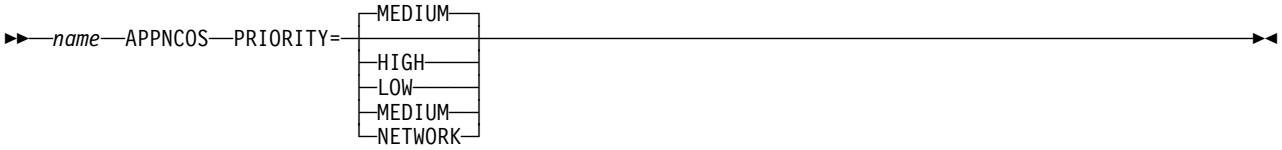
Details for implementing APPN classes of service are given in "APPN Class of Service" in the *VTAM Network Implementation Guide*.

See "IBM-Supplied APPN Classes of Service" on page 654 for descriptions of the seven IBM-supplied APPN classes of service, and "APPN Class-of-Service (COS) Definitions (COSAPPN)" on page 797 for samples of them.

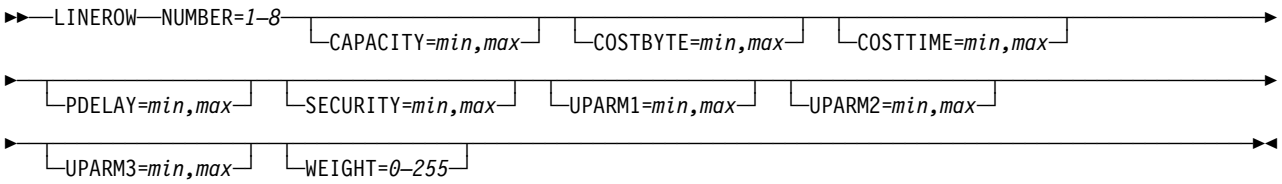
Full Syntax

The full syntax for the APPN class of service definition follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

APPNCOS



LINEROW

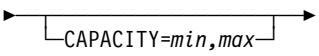


NODEROW

Operand Descriptions

The following section describes the operands you can code for the APPN class of service.

CAPACITY



statements:	LINEROW
dependency:	APPN only
range:	see values

specifies the range of the number of bits per second that the link can transmit data.

min and *max* have the same range of values:

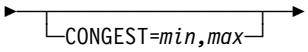
```
MINIMUM, 600, 1200, 2400, 4800, 9600, 14400
19200, 48000, 56000, 64000, 4M, 16M, MAXIMUM
```

The minimum default depends on the line row being defined:

Row	Minimum default
1	4M
2	56000
3	19200
4	9600
5	19200
6	9600
7	4800
8	MINIMUM

The maximum default is MAXIMUM for all line rows.

CONGEST



statements: NODEROW
dependency: APPN only
range: LOW–HIGH

indicates whether congestion is allowed for each node row.

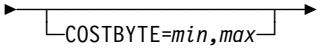
CONGEST allows you to indicate the weighting preference for uncongested nodes, or to disallow the use of congested nodes.

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



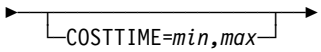
statements: LINEROW
dependency: APPN only
range: 0–255

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.

The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:

Row	Maximum default
1	0
2	0
3	0
4	0
5	0
6	128
7	196
8	255

COSTTIME



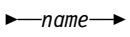
statements: LINEROW
dependency: APPN only
range: 0–255

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.

The minimum default is 0 for all line rows. The maximum default depends on the line row being defined:

Row	Maximum default
1	0
2	0
3	0
4	0
5	0
6	128
7	196
8	255

name



statements: APPNCOS
dependency: 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 (LINEROW)

statements: LINEROW
dependency: APPN only

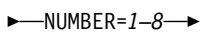
specifies a line row to be defined.

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)



statements: NODEROW
dependency: APPN only

specifies a node row to be defined.

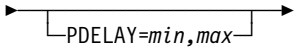
The default characteristic values of a given node row can differ depending on the number of the row being defined. So, an operand might have a different value for node row 7 than for node row 8. See the node row operand descriptions for defaults for each row number.

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



statements: LINEROW
dependency: 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).

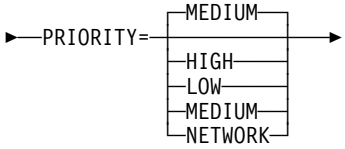
LONG
indicates a satellite delay (greater than 245.76 millisecs).

MAXIMUM
indicates a maximum delay.

The minimum default is MINIMUM for all line rows. The maximum default depends on the line row being defined:

Row	Maximum default
1	NEGLIGIB
2	TERRESTR
3	TERRESTR
4	TERRESTR
5	PACKET
6	PACKET
7	MAXIMUM
8	MAXIMUM

PRIORITY



statements: APPNCOS
dependency: APPN only

specifies one of the following transmission priorities for an APPN COS:

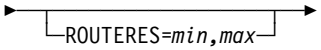
PRIORITY=LOW
is recommended for batch traffic

PRIORITY=MEDIUM
is recommended for normal traffic

PRIORITY=HIGH
is recommended for interactive traffic

PRIORITY=NETWORK
is used for APPN network services traffic, and is valid only for CPSVCMG and SNASVCMG.

ROUTERES



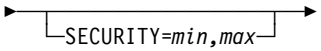
statements: NODEROW
dependency: APPN only
range: 0–255

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 (MVS, VM)



statements: LINEROW
dependency: 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.

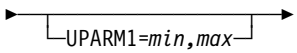
ENCRYPT
indicates link encryption.

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

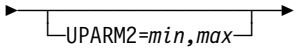


statements: LINEROW
dependency: APPN only
range: 0–255

specifies a range for a user-defined value. The user determines the meaning of this value.

Refer to Appendix H, “Forcing an APPN Route in a VTAM Network” in the *VTAM Network Implementation Guide* for an example of using the UPARM operands.

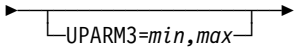
UPARM2



statements: LINEROW
dependency: APPN only
range: 0–255

specifies a range for a user-defined value. The user determines the meaning of this value.

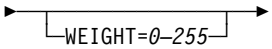
UPARM3



statements: LINEROW
dependency: APPN only
range: 0–255

specifies a range for a user-defined value. The user determines the meaning of this value.

WEIGHT (LINEROW)



statements: LINEROW
dependency: APPN only
range: 0–255

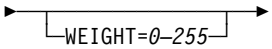
specifies the relative weight for each line row. Low weights indicate more desirable choices.

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.

The default depends on the line row being defined:

Row	Default
1	30
2	60
3	90
4	120
5	150
6	180
7	210
8	240

WEIGHT (NODEROW)



statements: NODEROW
dependency: 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, the value 150 is 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. 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 set of defaults 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. See "APPN Class-of-Service (COS) Definitions (COSAPPN)" on page 797 for samples.

MVS 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.

VM The IBM-supplied classes of service are shipped with the name COSAPPN VTAMLST on the LOCALSAM DISK (2c2).

VSE The IBM-supplied classes of service are shipped with the name COSAPPN in the VSE/VTAM production sublibrary.

Refer to "What are the IBM-supplied default classes of service?" in the *VTAM Network Implementation Guide* for more details.

#BATCH
An APPN class of service for LU-LU sessions that specifies a general batch-oriented class of service that uses low transmission priority, and for which high bandwidth and low cost are considered more important than short delay.

#BATCHSC
An APPN class of service for LU-LU sessions that specifies a general batch-oriented class of service 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 class of service for LU-LU sessions that provides connectivity at medium transmission priority.

CPSVCMG
An APPN class of service for CP-CP sessions that is used for network flows. It provides connectivity at network transmission priority.

#INTER
An APPN class of service for LU-LU sessions that specifies a general, interactive class of services that uses high transmission priority, and for which short delay is considered more important than high bandwidth and lost cost.

#INTERSC
An APPN class of service for a LU-LU session that specifies a general, interactive class of services 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 class of service for LU-LU CNOS sessions that provides connectivity at network transmission priority.

CPS Retry Table (VSE)

With X.21 short-hold mode/multiple port sharing (SHM/MPS), VTAM provides support for multiple physical units (PUs) over a limited number of lines. To share these lines, switched calls are dropped during inactive periods and are reestablished when data traffic starts. Therefore, a large number of calls can be attempted into the X.21 network.

X.21 call progress signal (CPS) retry allows X.21 network users to retry **non-retriable** call progress signals which are appropriate to their Post Telephone and Telegraph Administration (PTT) system by specifying the appropriate information in a CPS retry table.

This section discusses:

- Coding CPS retry routines
- Installing and changing the CPS retry table
- Coding the CPS retry table macroinstructions.

IBM supplies a sample CPS retry table on page 812.

Table 74. CPS Retry Table Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	CPSTAB	START	658	R			
name	CPSENTRY	CPSCODE	657	R			
		CPSRCNT	657			5	
		CPSRDLY	657			5	
name	CPSTAB	STOP	658	R			

Legend:

R Required operand.

Coding Notes

CPSTAB Macroinstruction: The CPSTAB macroinstruction defines the beginning and end of the CPS retry table.

CPSENTRY Macroinstruction: The CPSENTRY macroinstruction defines each call progress signal and its retry count and retry delay.

Installing and Changing a CPS Retry Table

The CPS retry table should be link-edited as a non-executable module into a private definition library referenced in the VTAM start procedure. VTAM loads the resulting load module during initialization processing.

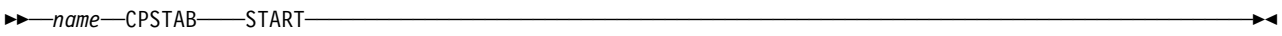
You specify the CPS number, its corresponding wait interval, and the respective number of retries in the CPS retry table using the CPSTAB start option.

The MODIFY TABLE command lets you dynamically replace a CPS retry table.

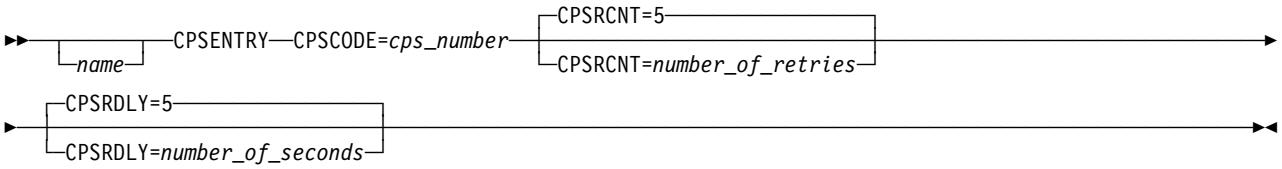
Full Syntax

The full syntax for the CPS retry table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

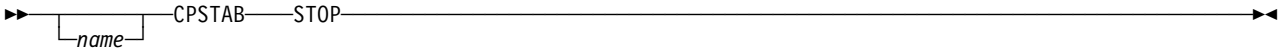
CPSTAB



CPSENTRY



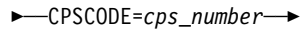
CPSTAB



Operand Descriptions

The following section describes the operands you can code for the CPS retry table.

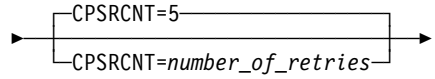
CPSCODE



macroinstructions: CPSENTRY
dependency: none

specifies the two-byte EBCDIC value that represents the call progress signal.

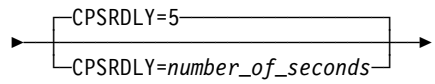
CPSRCNT



macroinstructions: CPSENTRY
dependency: none
range: 0–999

specifies the unsigned decimal value that represents the number of retries to attempt for this CPS.

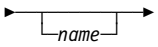
CPSRDLY



macroinstructions: CPSENTRY
dependency: none
range: 0–999

specifies the unsigned decimal value that represents the delay (in seconds) before VTAM should attempt a retry on this CPS.

name



macroinstructions: CPSTAB (STOP), CPSENTRY

macroinstructions: CPSTAB (START)

On CPSTAB (START): specifies a required 1–8 character name of the CPS retry table.

You can choose one of the following CPS tables:

- ISTCPS10, which contains the IBM-supplied call progress signals and their appropriate retry limits and retry delays.
- A user-defined table that contains call progress signals with retry limits and delays. If a **non-retriable** signal occurs that is not found in the table, it is **not** retried.

  Note: The table can also be used for **retriable** signals if the RETRYTO is 0 (zero) on the *LINE Statement for SDLC switched line*. In this case, if the **retriable** signal is not found, it is immediately retried.

On CPSTAB (STOP), CPSENTRY: specifies an optional 1–8 character name of the end of the table or a CPS entry.

START

macroinstructions: CPSTAB
dependency: none

indicates the beginning of the CPS table.

STOP

macroinstructions: CPSTAB
dependency: none

specifies the end of the CPS table.

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 714 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	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	INTAB					
name	LOGCHAR	APPLID SEQNCE	663 664	R		
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 that the logon sequences for the logon messages are from the most restrictive (greatest number of characters) to the least restrictive (fewest number of characters). An example of this follows:

```
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. Another way of stating this is that, 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 "Establishing and Controlling SNA Sessions" in the *VTAM 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."

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 "Logon-Interpret Routine Requirements" in *VTAM 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. **MVS** 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.

   MVS Put the user-written routines in a private call library.

   VM Refer to "Installing Tables and Modules in VM" in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit this table.

   VSE Put the routines in a private definition library referenced in the VTAM start procedure.

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.

   MVS Link-edit the resulting module as a non-executable module into a library identified by the SYS1.VTAMLIB DD statement.

   VM Refer to "Installing Tables and Modules in VM" in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit this table.

   VSE Link-edit the module into a private definition library referenced in the VTAM start procedure.

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 "Logon and Logoff Requests from Dependent Logical Units" in the *VTAM 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*. For linkage editor requirements, refer to *MVS/ESA Linkage Editor and Loader* or the *VM/SP Installation Guide*.

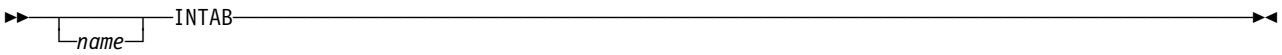
To change the name of an installed interpret table, do the following steps:

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 "Using the VARY ACT,UPDATE Technique" in the *VTAM 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.

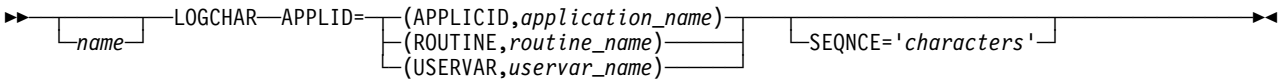
Full Syntax

The full syntax for the interpret table follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

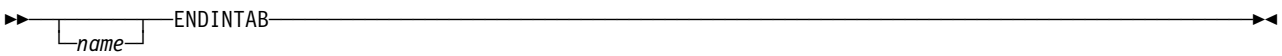
INTAB



LOGCHAR



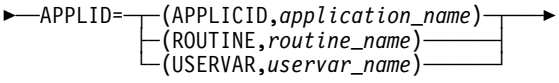
ENDINTAB



Operand Descriptions

The following section describes the operands you can code for the interpret table.

APPLID



macroinstructions: LOGCHAR
dependency: 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

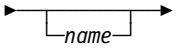
allowed to change. The resource name of the network-qualified name is considered generic and can undergo USERVAR translation.

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 "Logon-Interpret Routine Requirements" in *VTAM Customization* for details on coding logon-interpret routines.

APPLID=(USERVAR,*uservar_name***)**
is the same as specifying APPLICID.

name



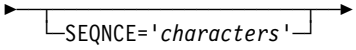
macroinstructions: INTAB, LOGCHAR, ENDINTAB
dependency: none

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.

SEQNCE



macroinstructions: LOGCHAR
dependency: 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.

MVS 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) (MVS, VSE)” on page 728 for the macroinstructions and operands that make up the default logon mode table.

VM The listing beginning on page 745 shows the macroinstructions that make up the default VM logon mode table. This table contains logon mode entries for use with VSCS and the NetView program.

IBM provides additional tables for VM. These, too, are shown in the listing on page 745. One of these (ISTTABLE) contains entries for terminals that might not be using VSCS (for example, terminals that need to log on to TSO). In addition, IBM supplies tables for use with the NetView program and the terminal access facility (TAF). See “Coding Logon Mode Tables for VSCS Devices” on page 630 for information on special logon mode table coding needed for VSCS.

VSE The listing beginning on page 728 shows the macroinstructions that make up the VSE default logon mode table.

For your information, the code released to you as the VTAM licensed program includes the source language for this table.

MVS The source language is located in SYS1.ASAMPLIB.

VM Refer to “Installing Tables and Modules in VM” in the *VTAM Network Implementation Guide* for information on where the source language is located.

VSE The source language is in the same library where VTAM definitions are located.

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	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	MODETAB					
name	MODEENT	APPNCOS	669			APPN only
		COMPRES	670		SYSTEM	MVS
		COMPROT	670		0	
		COS	671			
		DCODE	671			MVS
		ENCR	671		0	MVS data encryption
		FMPROF	672		0	
		LANG	672		0	MVS
		LOGMODE	674			
		PRIPROT	674		0	
		PSERVIC	674		0	
		PSNDPAC	674		0	
		RUSIZES	675			
		SECPROT	676		0	
		SRCVPAC	676		0	
		SSNDPAC	677		0	
		TSPROF	677		0	
		TYPE	677		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 “Session Parameter Fields (BIND Image)” in *VTAM 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.

VM For information on the coding restrictions of logon mode tables for VSCS devices, refer to the *VTAM Network Implementation Guide*.

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

MVS Each logon mode table must be assembled and link-edited into a library associated with the VTAMLIB DD statement.

VM Refer to "Installing Tables and Modules in VM" in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit this table.

VSE Assemble and link-edit the logon mode table into a private definition library referenced in the VTAM start procedure.

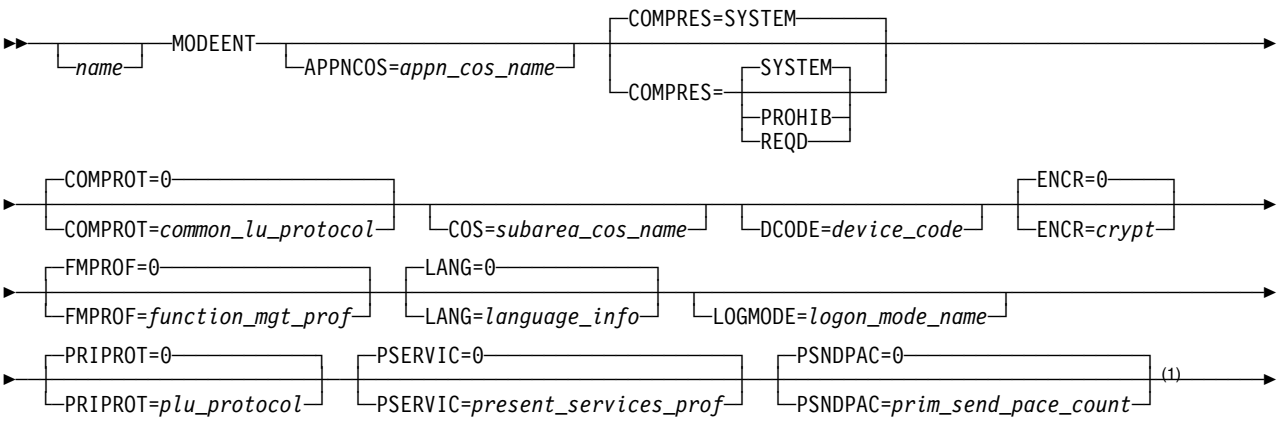
With the MODIFY TABLE command, the user can refresh or reload the logon mode table without deactivating the major nodes that should use the new table. Refer to "MODIFY TABLE Command" in *VTAM Operation* for information on using the MODIFY TABLE command.

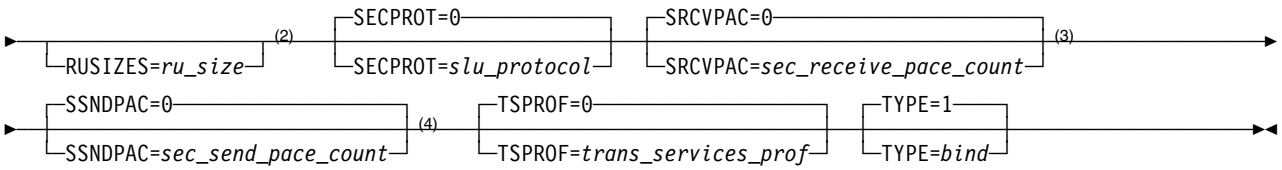
Full Syntax

The full syntax for the logon mode table follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

MODETAB

MODEENT





Notes:

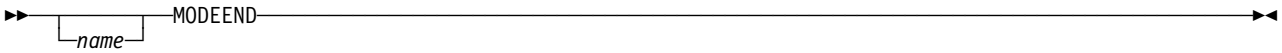
1 This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

2 This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

3 This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

4 This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

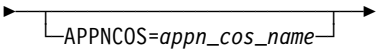
MODEEND



Operand Descriptions

The following section describes the operands you can code for the logon mode table.

APPNCOS



macroinstructions: MODEENT
dependency: 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.

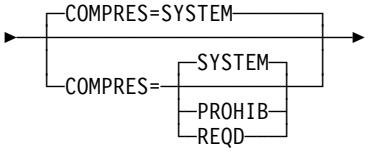
name can be 1–8 characters.

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.

COMPRES (MVS)



macroinstructions: MODEENT
dependency: none

specifies how the negotiation of data compression usage should be handled. This operand allows data-compression override specifications to be associated with a logmode 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 logmode.

COMPRES=REQD
specifies that override data should explicitly request data compression, regardless of link characteristics along the path, for sessions using this logmode, if both session partners can compress data.

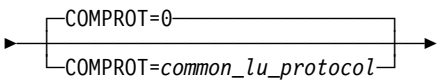
REQD causes compression to be specified only when you specify one of the following:

- 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 logmode, 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.

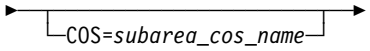
COMPROT



macroinstructions: MODEENT
dependency: 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

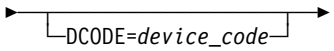


macroinstructions: MODEENT
dependency: subarea only

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 (MVS)



macroinstructions: MODEENT
dependency: none

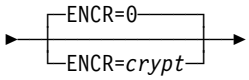
specifies the LU device code, and indicates to application programs whether a teletypewriter exchange service (TWX) or a World Trade teletypewriter (WTTY) device is a keyboard and printer or a keyboard and display terminal.

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 (MVS)



macroinstructions: MODEENT
dependency: 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

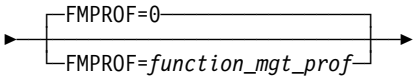
`00..` No private cryptography

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 Integrated Cryptographic Service Facility (ICSF/MVS), the IBM Programmed Cryptographic Facility program product (PCF), or the IBM Cryptographic Unit Support program product (CUSP).

FMPROF

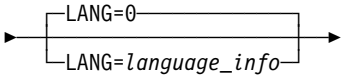


macroinstructions: MODEENT
dependency: none

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 “Function Management Profile” in *VTAM Programming*.

LANG (MVS)



macroinstructions: MODEENT
dependency: none
range: 0–255. or
X'0–FF'

defines the language information that will be passed to the VTAM application program. Refer to “Node Initialization Block (NIB)” in *VTAM 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 X'*number*'.

MVS 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 Appendix A, “TSO/VTAM ” in the *VTAM 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.

MVS 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 "LANG Operand of the MODEENT Macroinstruction" in the *VTAM Network Implementation Guide*.

For a list of valid values to use for this field, refer to "Node Initialization Block (NIB)" in *VTAM 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 714 for more information.

The meanings of the LANG values follow:

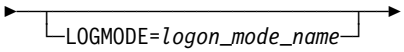
Bit 0123 4567	Description
0...	The character set is identified from the code specified in bits 1–7. The device is assumed incapable of double-byte character set (DBCS) processing.
1...	A query command is sent to the device to determine the character set and DBCS capability. If the character set cannot 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.

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.

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.

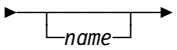
LOGMODE



macroinstructions: MODEENT
dependency: none

specifies the logon mode name to be used as a key for the session parameters in this table entry. If duplicate names appear in the table, the first occurrence of the name is used. If you omit LOGMODE, 8 blanks are used.

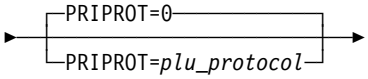
name



macroinstructions: MODETAB, MODEENT, MODEEND
dependency: none

specifies optional names for the MODETAB, MODEENT, and MODEEND macroinstructions.

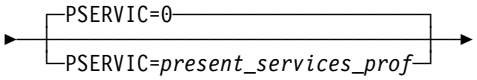
PRIPROT



macroinstructions: MODEENT
dependency: 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.

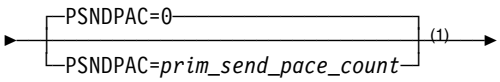
PSERVIC



macroinstructions: MODEENT
dependency: 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 “Logical Unit Presentation Services Profile” and “Logical Unit Presentation Services Usage Field” in *VTAM Programming*.

PSNDPAC

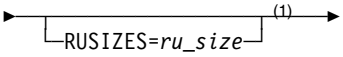


Note:
[1] This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

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 “Pacing Count” in *VTAM Programming* for more information.

RUSIZES



Note:
[1] This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

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' (65535 in decimal).
- If the BIND is not an extended BIND, the default is the value of the MAXLURU start option.

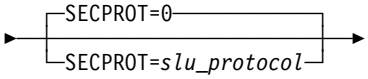
For APPC sessions, if the RUSIZES value is zero or less than 256, the value is overridden and set to 256.

Table 77 (Page 1 of 2). 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

Table 77 (Page 2 of 2). Exponent and Mantissa Values for Each RU size								
Exp (n)	**Mantissa (m)**							
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
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

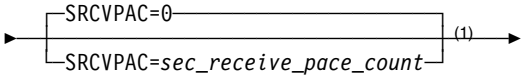
SECPROT



macroinstructions: MODEENT
dependency: 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

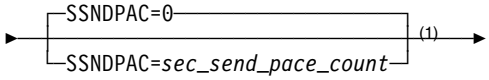


Note:
[1] This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

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 “Pacing Count” in *VTAM Programming* for more information.

SSNDPAC

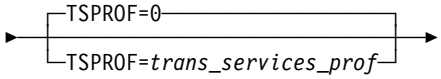


Note:
[1] This operand applies only when the session partner of the LU for which this logon mode table applies is an independent LU.

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 “Pacing Count” in *VTAM Programming* for more information.

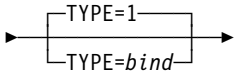
TSPROF



macroinstructions: MODEENT
dependency: 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 “Transmission Services Profile” in *VTAM Programming*.

TYPE



macroinstructions: MODEENT
dependency: 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 “BIND Request” in *VTAM 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 Table

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	FLDTAB					
name	FLDENT	MESSAGE	681	R		
		HARDCOPY	681		YES	
		TIME	681		30	
		VARTEXT	682		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

MVS Store the set of macroinstructions as a member of SYS1.VTAMLST. Use the member name as the name of the table.

VM Store the macroinstructions with a filetype of VTAMLST and use the filename as the name of the table.

VSE Store the macroinstructions in the private definition library referenced in the VTAM start procedure.

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 "MODIFY TABLE Command" in *VTAM 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 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.

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. 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.

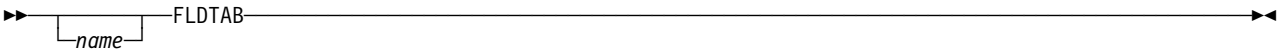
The message at Time 35 would be displayed since it was issued after the 30 second time-span.

```
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
```

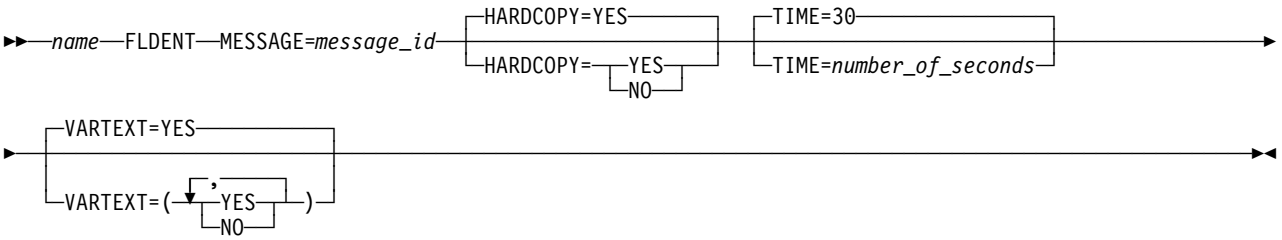
Full Syntax

The full syntax for the message-flooding prevention table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

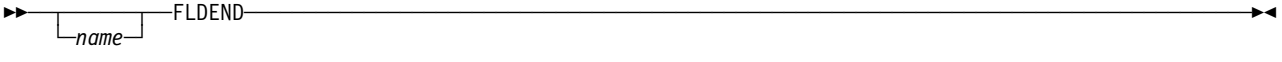
FLDTAB



FLDENT



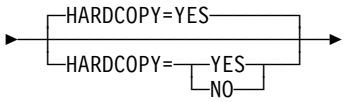
FLDEND



Operand Descriptions

The following section describes the operands you can code for the message-flooding prevention table.

HARDCOPY



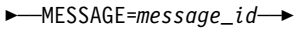
macroinstruction: FLDENT

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.

MESSAGE

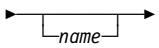


macroinstruction: FLDENT

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



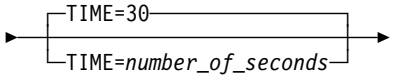
macroinstructions: FLDTAB, FLDEND *macroinstructions:* FLDENT

specifies the name of the macroinstruction being defined.

For FLDTAB and FLDEND: *name* is optional and has no function. MVS 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.

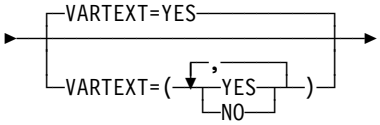
TIME



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



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.

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 Chapter 2, "Major Nodes" on page 19.

Table 79. Model Name Table Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Sift Effect	Default	Restrictions, Dependencies
name	MDLTAB						
name	MDLENT	MODEL	684				
name	MDLPLU	PLU MODEL	685 684	R			

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

MVS Store the model name table macroinstructions in SYS1.VTAMLST, and use the member name as the name of the table.

VM Store the macroinstructions with a filetype of VTAMLST and use the filename as the name of the table.

VSE Store the macroinstructions in the private definition library referenced in the VTAM start procedure.

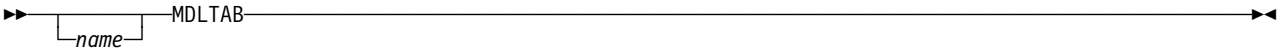
No assembly or link-edit is required to install the table. The table will be dynamically built upon the first activation of any resource that has a defined association to the table. When the last resource using it is deactivated, the table is deleted and its storage is freed.

IBM does not supply a default model name table.

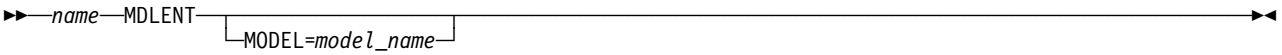
Full Syntax

The full syntax for the model name table follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

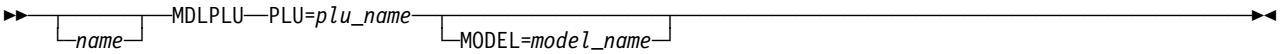
MDLTAB



MDLENT



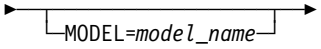
MDLPLU



Operand Descriptions

The following section describes the operands you can code for the model name table.

MODEL



macroinstructions: MDLENT, MDLPLU

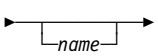
specifies the 1–8 character model name expected by the subsystem for the terminal.

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

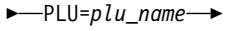


macroinstructions: MDLTAB, MDLPLU

macroinstructions: MDLENT

specifies the 1–8 character name for the macroinstructions.

PLU



macroinstructions: MDLPLU

specifies the 1–8 character name of the PLU as it is known in the SLU's network. cms

Session Awareness (SAW) Data Filter (MVS)

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 14 for information on assembling modules.

You can modify ISTMGC10 or replace it with one of your own using the MODIFY TABLE command. Refer to “MODIFY TABLE Command” in *VTAM Operation* for more information on the MODIFY TABLE command.

Table 80. Session Awareness Data Filter Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	KEEPMEM	START-STOP	689	R		
name	KCLASS	SAW	688	R	YES	
name	MAPSESS	KCLASS PRINET PRI SEC SECNET	688 688 688 689 689	R		
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:

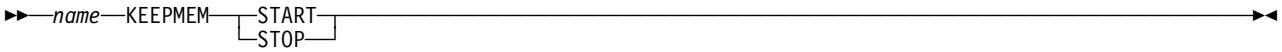
1. The maximum number of MAPSESS macroinstructions you can code is 2000.
2. If the PLU has the host CDRM name coded, PRI=VTAM must be specified. PRI=VTAM is used to obtain saw data before the activation of the host CDRM major node.

END Macroinstruction: The END macroinstruction indicates the end of a SAW data filter.

Full Syntax

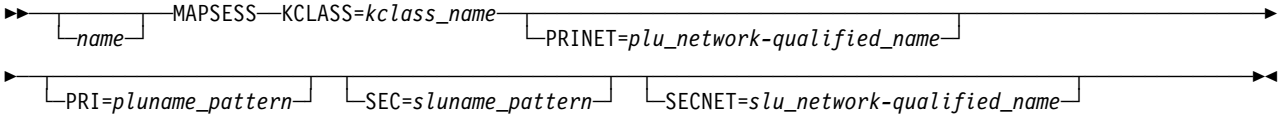
The full syntax for the session awareness data filter follows. The syntax for each operand is repeated in the “Operand Descriptions” section.

KEEPMEM

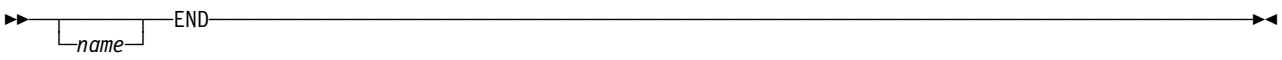


KCLASS

MAPSESS



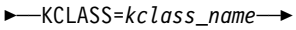
END



Operand Descriptions

The following section describes the operands you can code for the saw data filter.

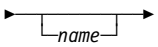
KCLASS



macroinstructions: MAPSESS
dependency: 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

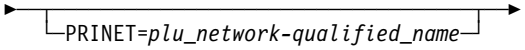


macroinstructions: MAPSESS, END

macroinstructions: KEEPMEM, KCLASS

specifies 1–8 character names for the macroinstructions.

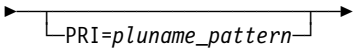
PRINET



macroinstructions: MAPSESS
dependency: none

specifies a network-qualified name for the PLU.

PRI



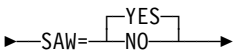
macroinstructions: MAPSESS
dependency: none

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.
- * matches anything.
- T3??7 matches T3AB7 or T3CD7. It does not match T3A7.

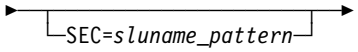
SAW



macroinstructions: KCLASS
dependency: none

specifies whether to pass SAW data over the CNM interface.

SEC

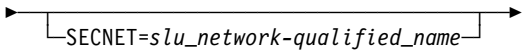


macroinstructions: MAPSESS
dependency: none

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



macroinstructions: MAPSESS
dependency: none

specifies a network-qualified name for the SLU.

START-STOP

macroinstructions: KEEPMEM
dependency: none

specifies whether this macroinstruction starts or stops a set of session awareness data filter definition statements.

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 *VTAM Messages and Codes*.

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) (MVS)" on page 766 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) (MVS)" on page 770 for an example of an operation-level USS table.

IBM supplies default USS tables with the VTAM product. MVS The source language for these tables is contained in SYS1.ASAMPLIB. VM Refer to "Installing Tables and Modules in VM" in the *VTAM Network Implementation Guide* for information on where the source language is located. VSE The source language is in the same library where VTAM definitions are located. Table 81 describes the default tables.

Table 81. 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 – UNDIAL VM,VSE • USSMSGxx (where xx equals 1–9 and 11–14) that are sent to a terminal operator. Refer to *VTAM Messages and Codes* 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 *VTAM Operation*. • VTAM operator messages listed in the *VTAM Messages and Codes*.
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 "MODIFY TABLE Command" in *VTAM 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 709.

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 712.

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 82 through 85 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 82.

Table 82. VTAM Messages Search Order for VTAM Messages

Terminal Operator	VTAM Operator
MVS Message Service[1] on page 693	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 "National Language Support for End-User USS Messages and Commands" in the *VTAM 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 83.

Table 83. 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 84.

Table 84. 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
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 85.

Table 85. 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

MVS Each USS table must be assembled and link-edited as a non-executable module into a library associated with the SYS1.VTAMLIB DD statement.

VM Refer to "Installing Tables and Modules in VM" in the *VTAM Network Implementation Guide* for information on how to assemble and link-edit this table.

VSE To assemble and link-edit this table, use a private definition library referenced in the VTAM start procedure.

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) (MVS)" on page 770.

Table 86 shows the USS macroinstructions, their associated operands, and where the operands are described.

Table 86 (Page 1 of 2). USS Table Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
name	USSTAB	DATEDLM	698		/	FORMAT; V4
		DATEFRM	699		MDY	FORMAT; DATEDLM; V4
		FORMAT	700		OLD	
		TABLE	707			
		TIMEDLM	708			FORMAT; V4
name	USSCMD	CMD	698	R		

Table 86 (Page 2 of 2). USS Table Macroinstructions and Operands

Name	Macroinstruction	Operands	Page	Req	Default	Restrictions, Dependencies
		FORMAT	701		PL1	
		REP	705			
name	USSPARM	PARM	704	R		
		DEFAULT	699			
		REP	705			
		VALUE	708			
name	USSMSG	BUFFER	697	R		Terminal operator; TEXT; OPT; SUPP
		MSG	702	R		
		TEXT	707	R		Required for terminal operator BUFFER
		DESC	700			VTAM operator
		MCSFLAG	702			VTAM operator
		MSGLVL	702		START	VTAM operator
		OPT	704		BLKSUP	terminal operator
					NOBLKSUP	VTAM operator
		ROUTCDE	706			VTAM operator
		SUPP	706		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).

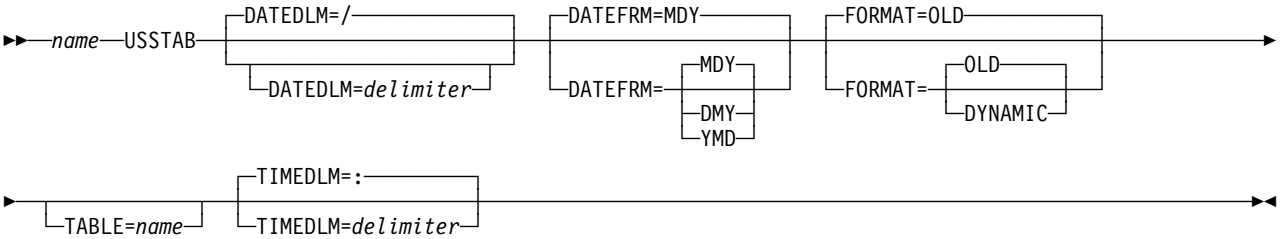
To determine under what circumstances these messages are issued, refer to Chapter 8, “USS Messages” in *VTAM Messages and Codes*.

USSEND Macroinstruction: The USSEND macroinstruction delimits the end of the USS table.

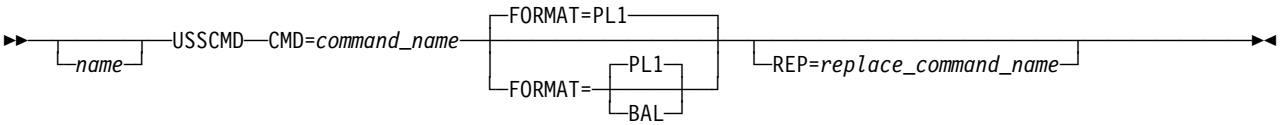
Full Syntax

The full syntax for the USS table follows. The syntax for each operand is repeated in the "Operand Descriptions" section.

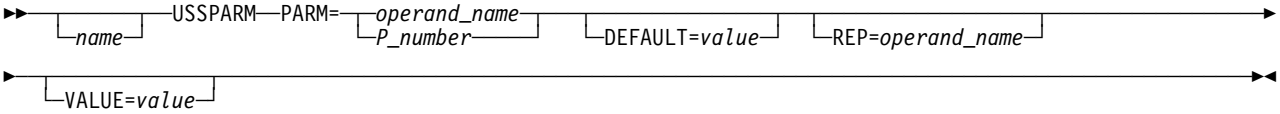
USSTAB



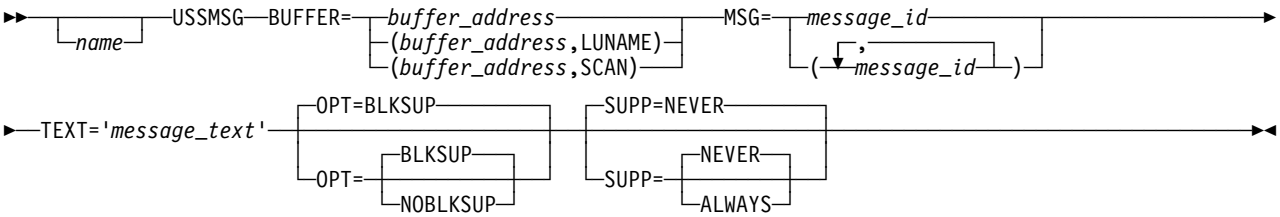
USSCMD



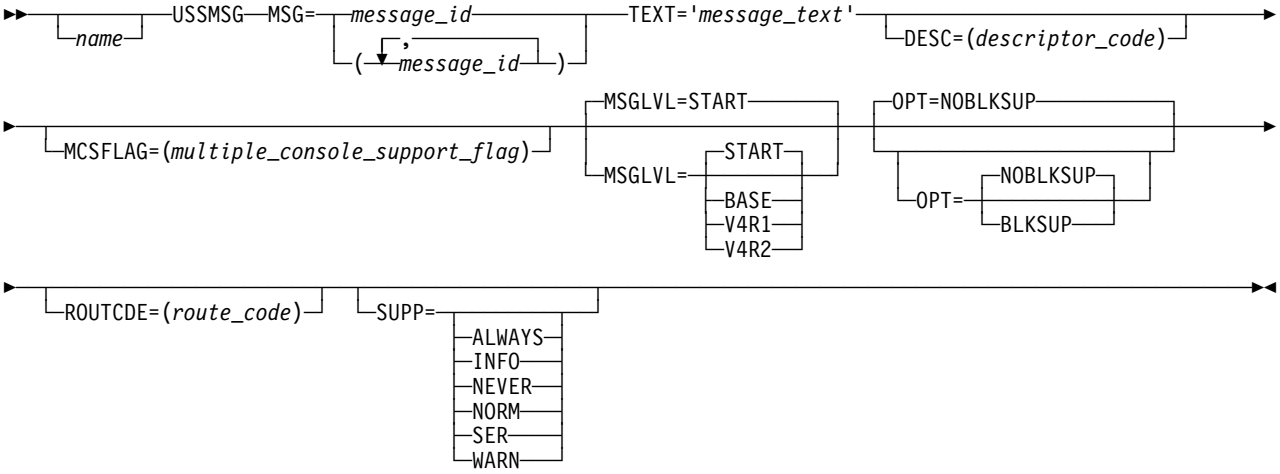
USSPARM



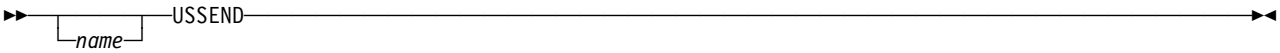
USSMSG (Terminal Operator Messages)



USSMSG (VTAM Operator Messages)



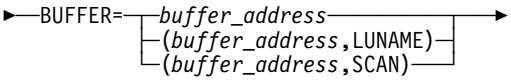
USSEND



Operand Descriptions

The following section describes the operands you can code for the USS table.

BUFFER



macroinstructions: USSMSG
dependency: 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 3.

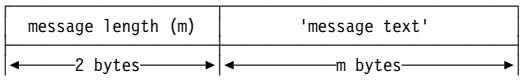


Figure 3. USS Message Layout in Storage

The message text defined in the storage area must follow the rules listed in “Syntax Rules for Messages” on page 710.

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 87 on page 698, 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 87. 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*.
@@@@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.

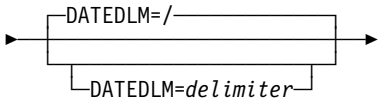
CMD

macroinstructions: USSCMD
dependency: 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



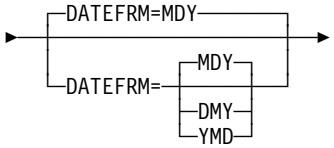
macroinstructions: USSTAB
dependency: FORMAT=DYNAMIC; V4 and later versions

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



macroinstructions: USSTAB
dependency: 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.

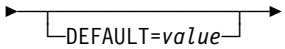
DATEFRM=DMY
specifies the day, followed by month, followed by year as dd_mm_yy, where an underscore (_) is the delimiter specified on the DATEDLM operand.

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



macroinstructions: USSPARM
dependency: 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.

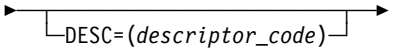
The default value (*value*) must follow the rules for coding assembler DC statements for character (C-type) constants. For example, if you want the following character string for *value*

```
'DUMP'
```

code

```
DEFAULT=''DUMP''
```

DESC



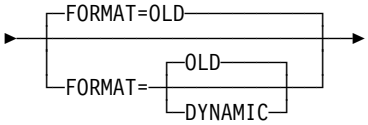
macroinstructions: USSMSG
dependency: VTAM operator messages only

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)



macroinstructions: USSTAB
dependency: none

specifies how the USS table will be formatted.

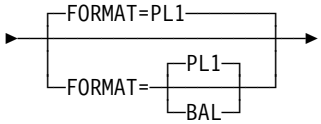
FORMAT=DYNAMIC
specifies that the MODIFY TABLE command can be used to dynamically replace the USS table. Also, use FORMAT=DYNAMIC to permit terminal operators to specify the USS table using the LANGTAB operand on a USS command.

Specifying FORMAT=DYNAMIC can significantly reduce the CPU time required to search for VTAM operator messages. Refer to “MODIFY TABLE Command” in *VTAM 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.

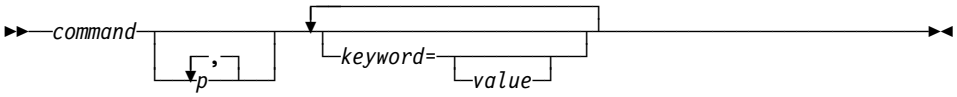
FORMAT (USSCMD)



macroinstructions: USSCMD
dependency: 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.



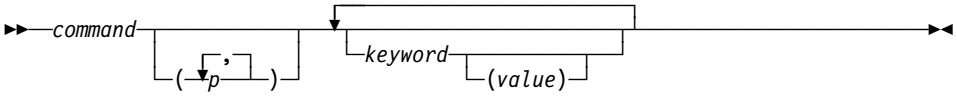
command identifies the command. It is followed by one or more blanks.

p used to specify one or more positional operand. Positional operands are entered in the format P*n*, where *n* is the position number of the operand. Each operand (unless it is the last in the command) is followed by a comma. Positional operands must appear before any keyword operands.

keyword used to specify keyword operand associated with the command. Each operand (unless it is the last in a command) is followed by a comma.

value is the value assigned to a keyword operand.

FORMAT=PL1
specifies the user-defined command specified on this USSCMD macroinstruction is in PL/1 programming syntax.



command identifies the command. It is followed by one or more blanks or by a left parenthesis (that is, positional operands).

p specifies one or more positional operands. Positional operands are entered in the format P*n*, where *n* is the position number. If positional operands are used, the parentheses must be coded.

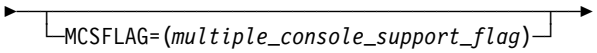
keyword is used to enter each operand parameter. Each operand must be followed by one or more blanks or by a value enclosed in parentheses.

value is the value assigned to a keyword operand.

See “Syntax Rules for Terminal Operator Commands” on page 713 for a description of the syntax, the input character set, and BAL and PL/1 value restrictions.

Note: If the USS table is to be searched twice (REP=VERB coded), this operand is ignored on the second pass of USS processing.

MCSFLAG



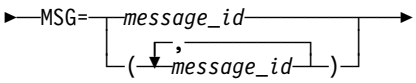
macroinstructions: USSMSG
dependency: VTAM operator messages only

specifies the multiple console support flags to be used for this message. This operand is coded exactly like, and has the same meaning as, the MCSFLAG parameter of the WTO or WTOR macroinstruction. For information on how to code this operand, refer to *MVS Supervisor Services and Macros*.

If the MCSFLAG operand is omitted or is specified with a null value (MCSFLAG=), VTAM uses the MCSFLAG value specified for this message ID in the next USS table searched.

If MCSFLAG is not coded in any of the USS tables that are searched, its value will default to zero.

MSG



macroinstructions: USSMSG
dependency: none

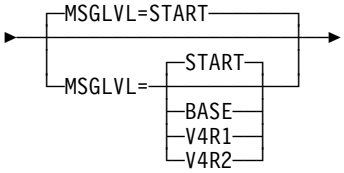
specifies which message or messages will be defined by this macroinstruction.

Terminal operator messages: Enter decimal integers in the range 0–14. The numbers 0–14 correspond to the USS messages with message IDs of USSMSG00 through USSMSG14, respectively. Refer to Chapter 8, “USS Messages” in *VTAM Messages and Codes* for a description of USSMSG00 through USSMSG14.

Note: USSMSG00 and USSMSG10 are not defined in the IBM-supplied USS table. If you do not define these messages, no messages are sent for these conditions.

VTAM operator messages: To define a VTAM message, specify a valid VTAM message ID. Enter the entire message ID in its correct format (for example, MSG=IST123I).

MSGLVL



macroinstructions: USSMSG
dependency: 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 “Message Text for BASE and

Version 4 Messages" in *VTAM Messages and Codes*. 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 "Message Text for BASE and Version 4 Messages" in *VTAM Messages and Codes* 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.

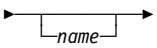
MSGLVL=START
specifies that for the message number indicated on the MSG operand, issue the version of the message specified on the MSGLEVEL start option. START is the default for only those messages supported by the MSGLVL operand.

MSGLVL=V4R1
specifies that for the message number indicated on the MSG operand, issue the V4R1 message or messages, regardless of the MSGLVL value specified on the MSGLEVEL start option. Refer to "Message Text for BASE and V4R1 Messages" in *VTAM Messages and Codes* 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 "Message Text for BASE and V4R2 Messages" in *VTAM Messages and Codes* for a list of messages supported by the MSGLVL operand.

name

macroinstruction: USSTAB



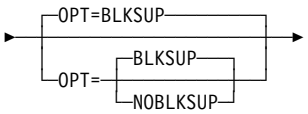
macroinstruction: USSCMD, USSPARM
USSMSG, USSEND

For USSCMD, USSPARM, USSMSG, USSEND: specifies the name assigned to the macroinstruction.

For USSTAB specifies the required CSECT name for the USS table.

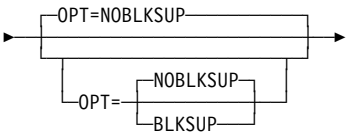
OPT

For Terminal Operator



macroinstructions: USSMSG
dependency: BUFFER

For VTAM Operator



macroinstructions: USSMSG
dependency: none

indicates if extraneous blanks are to be suppressed from the defined message.

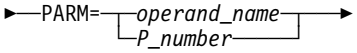
OPT=BLKSUP
specifies that extraneous blanks are to be suppressed from the message. Any sequence of two or more blanks is to be converted into a single blank.

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



macroinstructions: USSPARM
dependency: none

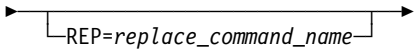
specifies a user-defined keyword or positional parameter that can be coded in the user-defined command identified by the USSCMD definition.

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 819 for an example of positional and keyword parameters.

REP (USSCMD)

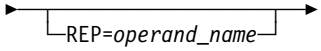


macroinstructions: USSCMD
dependency: none

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 819 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 819 for an example.

REP (USSPARM)



macroinstructions: USSPARM
dependency: 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 819).

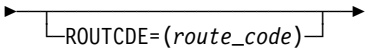
REP=VERB
indicates that the command is replaced with the value of the VALUE operand. The USS table is then searched a second time, matching this new command to the value of a CMD operand on a USSCMD macroinstruction.

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



macroinstructions: USSMSG
dependency: VTAM operator messages only

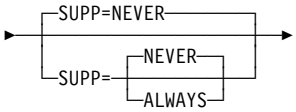
specifies the routing code to be used for this message. This operand is coded exactly like, and has the same meaning as, the ROUTCDE operand of the WTO or WTOR macroinstruction. For information on how to specify this operand, refer to *MVS Supervisor Services and Macros*.

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.

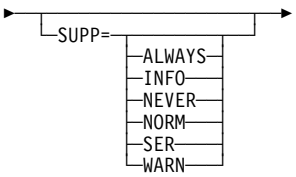
SUPP

Terminal Operator



macroinstruction: USSMSG
dependency: BUFFER

VTAM Operator



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 *VTAM Messages and Codes* 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.

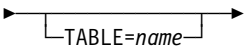
SUPP=SER
specifies that this message is to be in the serious message suppression class.

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

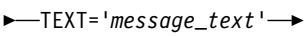


macroinstructions: USSTAB
dependency: none

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



macroinstructions: USSMSG
dependency: 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 710.

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 87 on page 698; 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
```

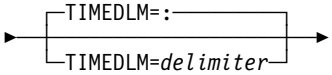
An example using date and time follows:

```
TEXT='IT IS @@@@TIME ON @@@@DATE'
```

produces a message like the following:

```
IT IS 10:14:51 ON 10/10/91
```

TIMEDLM

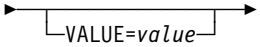


macroinstructions: USSTAB
dependency: FORMAT=DYNAMIC; V4 and later versions

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



macroinstructions: USSPARM
dependency: 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 819).

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 “Changing Message Characteristics” in *VTAM Messages and Codes*.
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 “National Language Support for End-User USS Messages and Commands” in the *VTAM 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 692 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 88 shows the VTAM message equivalent of USSMSG00 to USSMSG14.

Table 88 (Page 1 of 2). 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

Table 88 (Page 2 of 2). USS Messages and VTAM Message Equivalents

USS Message Number	VTAM Operator Message
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)'
  ```

  Appendix E, “Message Text for VTAM Operator Messages” in *VTAM Messages and Codes* 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 or NET *
VARY	NET

Note:

* VM This operand is optional.

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 714
- “LOGOFF Command” on page 717
- “IBMTEST Command” on page 720
- “UNDIAL Command (VM, VSE)” on page 722

VTAM converts each of these unformatted commands (except for UNDIAL for VM) into a field-formatted SNA request. VM For the UNDIAL command, VTAM issues a CP RESET command for the terminal.

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 89. LOGON Command and Operands

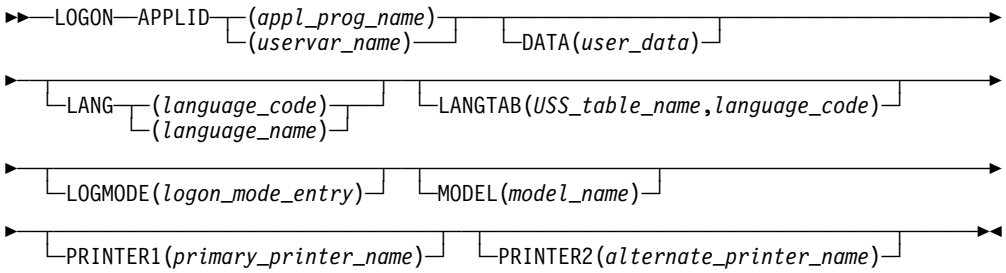
Command	Operands	Page	Req	Default	Restrictions, Dependencies
LOGON	APPLID(*appl_prog_name* or *uservar_name*)	715	R		
	DATA(*user_data*)	715			
	LANG(*language_code* or *language_name*)	715			MVS
	LANGTAB(*uss_table_name*,*language_code*)	716			
	LOGMODE(*logon_mode_entry*)	716			
	MODEL(*model_name*)	716			
	PRINTER1(*primary_printer_name*)	717			
	PRINTER2(*alternate_printer_name*)	717			

Legend:

R Required operand.

Full Syntax

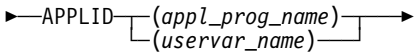
The full syntax for the LOGON command follows. The syntax for each operand is repeated in the “Operand Descriptions” section.



Operand Descriptions

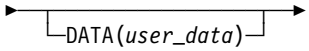
The following section describes the operands you can code for the LOGON command.

APPLID



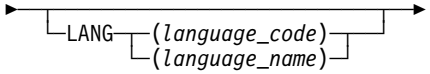
specifies the name of the application program or character string specified by the SEQNCE entry in the interpret table (USERVAR) with which a session is to be established. For more information on network regulations regarding name specification, see “APPL Definition Statement” on page 30.

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 (MVS)



dependency: 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.

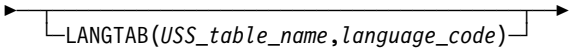
For non-MVS systems, the LANG operand is accepted, but ignored.

LANG(*language_code***)**
specifies a 3-character code that indicates a valid language. For a list of valid language code values, refer to "Node Initialization Block (NIB)" in *VTAM 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



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.

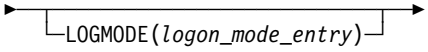
USS_table_name
specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

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 "Node Initialization Block (NIB)" in *VTAM Programming*.

Notes:

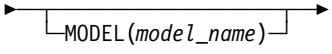
1. If you enter this operand without a USS table name, it removes the LANGTAB USS table (specified earlier) from the search hierarchy.
2. If you enter this operand without a language code, the application receives the language from the LANG operand of the MODEENT macroinstruction (specified by LOGMODE).

LOGMODE



specifies the logon mode entry that is used to select a set of session parameters for the session to be established.

MODEL

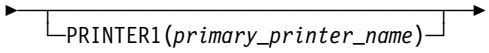


specifies the 1–8 character model name to be passed to the application program during logon processing.

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

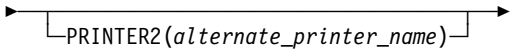


specifies the primary printer name to be passed to the application program during logon processing.

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



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

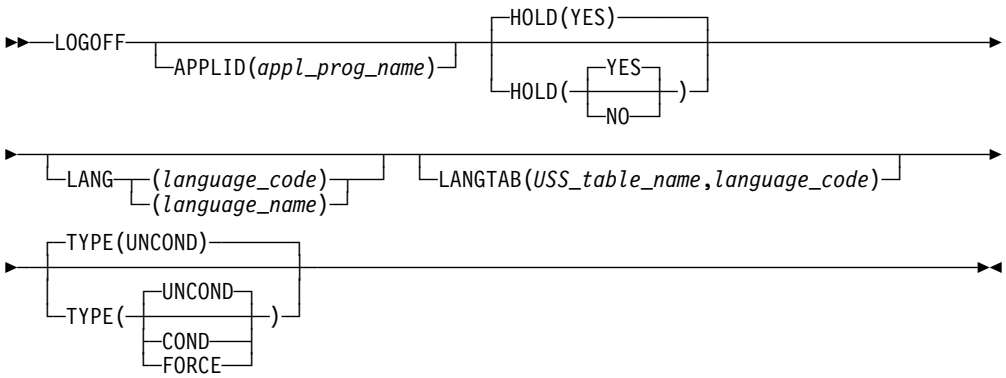
The LOGOFF command allows the terminal operator to request that a session with an application program be ended.

Table 90. LOGOFF Command and Operands

Command	Operands	Page	Req	Default	Restrictions, Dependencies
LOGOFF	APPLID(*appl_prog_name*)	718			
	HOLD(*YES* or *NO*)	718		YES	token ring
	LANG(*language_code* or *language_name*)	718			MVS Message Service
	LANGTAB(*uss_table_name*,*language_code*)	719			
	TYPE(COND or FORCE or UNCOND)	720		UNCOND	

Full Syntax

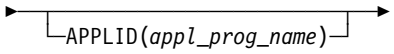
The full syntax for the LOGOFF command follows. The syntax for each operand is repeated in the “Operand Descriptions” section.



Operand Descriptions

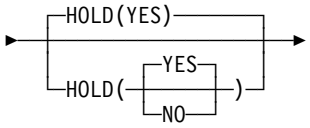
The following section describes the operands you can code for the LOGOFF command.

APPLID



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

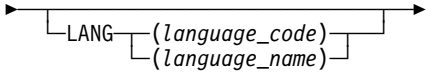


dependency: do not code for token ring

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 “Switched SDLC Connection” in the *VTAM 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



dependency: 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.

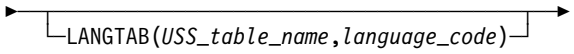
For non-MVS systems, the LANG operand is accepted, but ignored.

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 "Node Initialization Block (NIB)" in *VTAM Programming*.

For information on the MVS Message Service, see *MVS/ESA Planning: Operations* and the *MVS/ESA Application Development Guide: Assembler Language Programs*.

LANGTAB



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.

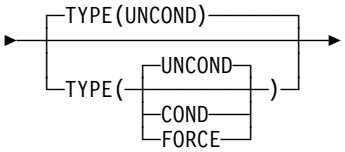
USS_table_name
specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

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 "Node Initialization Block (NIB)" in *VTAM 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



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 91. IBMTEST Command and Operands

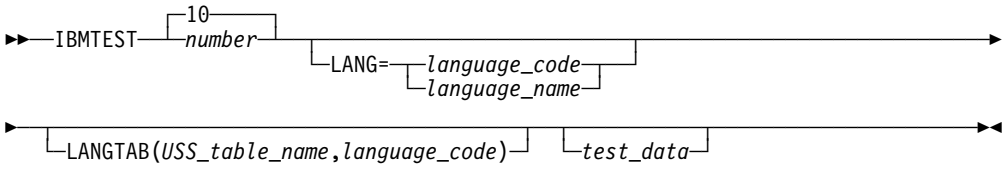
Command	Operands	Page	Req	Default	Restrictions, Dependencies
IBMTEST	*number*	722	R	10	
	LANG(*language_code* or *language_name*)	721			
	LANGTAB(*uss_table_name*,*language_code*)	721			
	test_data	722			

Legend:

R Required operand.

Full Syntax

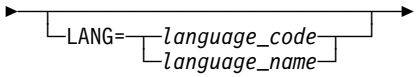
The full syntax for the IBMTEST command follows. The syntax for each operand is repeated in the "Operand Descriptions" section.



Operand Descriptions

The following section describes the operands you can code for the IBMTEST command.

LANG



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.

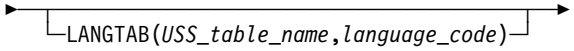
Note: If the MVS Message Service is not currently active, the LANG operand is ignored. For non-MVS systems, the LANG operand is accepted, but ignored.

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 "Node Initialization Block (NIB)" in *VTAM 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



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.

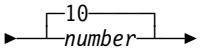
uss_table_name
specifies the USS table that VTAM will search first for USS messages and commands. VTAM uses this USS table until the terminal operator issues another USS command using the LANGTAB operand.

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 "Node Initialization Block (NIB)" in *VTAM 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).

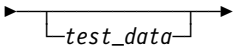
number



range: 1–255

specifies the number of times the test data should be returned to the terminal.

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.

UNDIAL Command (VM, VSE)

Terminal operators can use the UNDIAL command to request that their terminal be disconnected from the VTAM virtual machine and returned to the control of the VM Control Program.

You should issue the UNDIAL command only if you have previously issued the CP DIAL command to connect your terminal to the VTAM virtual machine. VTAM uses the CP RESET command to reset the terminal. This is accomplished with the SSCP-LU session flow. VTAM ignores the command if it does not apply to the terminal from which it is invoked.

Note that if a USS terminal operator table other than ISTINCDT is used for local terminals, the following must be added to the table to use the UNDIAL command:

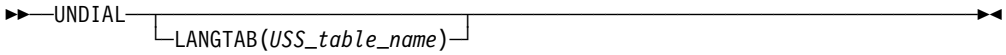
```
UNDIAL USSCMD   CMD=UNDIAL,FORMAT=PL1
```

Table 92. UNDIAL Command and Operand

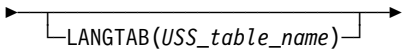
Command	Operands	Page	Req	Default	Restrictions, Dependencies
UNDIAL	LANGTAB(*uss_table_name*)	723			

Full Syntax

The full syntax for the UNDIAL command follows.



LANGTAB



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.

Appendixes

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 sample X.21 call progress signal (CPS) retry table for VSE
- The APPN class-of-service (COS) definitions (COSAPPN)
- The APPN transmission group (TG) profile definitions (IBMTGPS)
- The message flooding table (ISTMSFLD)

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 11 for correct coding formats, especially when continuing lines of code.

Note: “Coding Logon Mode Tables for VSCS Devices” on page 630 contains information about customizing the logon mode table in a VM environment.

Default Logon Mode Table (ISTINCLM) (MVS, VSE)

```
         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
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'020000000000185000007E00',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'020000000000185020507F00',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'03000000000018502B507F00',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'02000000000000C280C507F00',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'02000000000018500000 7E00',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'02000000000018500000 7E00',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'028000000000000000000300',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'020000000000000000000100',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'020000000000000000000200',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'020000000000185000007E00',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'020000000000185000007E00',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'020000000000185020507F00',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'020000000000185028507F00',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'028000000000000000000300',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'000000000000185020507F00',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'00800000000000000000000300',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
*                                                                   *
*********************************************************************
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'008000000000185000007E00',                    *
              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'008000000000185020507F00',                    *
              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'028000000000185000007E00',                    *
              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'028000000000185020507F00',                   *
               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'028000000000185000007E00',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'028000000000185020507F00',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)
*                                                                   *
*********************************************************************
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'008000000000185000007E00',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'008000000000185000007E00',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'028000000000185000007E00',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'028000000000185000007E00',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'8585',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 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 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'060200000000000000000300',   *
               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'060200000000000000102F00',   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
         MODEEND ,                    END OF DEFAULT TABLE ENTRIES
         END                        , END OF ISTINCLM
```

Default Logon Mode Tables (ISTINCLM) (VM)

```
         EJECT
ISTINCLM MODETAB
         EJECT
         TITLE 'INTERACT'
*********************************************************************
*                                                                   *
*              3767 INTERACTIVE                                     *
*                                                                   *
*********************************************************************
IBM3767  MODEENT LOGMODE=INTERACT,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',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
         TITLE 'VSCS3767'
*********************************************************************
*                                                                   *
*              VSCS 3767 KEYBOARD PRINTER                           *
*                                                                   *
*********************************************************************
VSCS3767 MODEENT LOGMODE=VSCS3767,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               PSERVIC=X'014000000000000000000000',                  *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'VSCS2741'
*********************************************************************
*                                                                   *
*              VSCS 2741 KEYBOARD PRINTER                           *
*                                                                   *
*********************************************************************
VSCS2741 MODEENT LOGMODE=VSCS2741,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               PSERVIC=X'014000000000000000000000',                  *
```

```
              APPNCOS=#CONNECT
         EJECT
         TITLE 'VSCSTWX'
*********************************************************************
*                                                                   *
*              VSCS TWX                                             *
*                                                                   *
*********************************************************************
VSCSTWX  MODEENT LOGMODE=VSCSTWX,FMPROF=X'03',TSPROF=X'03',            *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',             *
              PSERVIC=X'010000000000000000000000',                     *
              APPNCOS=#CONNECT
         TITLE 'BATCH'
*********************************************************************
*                                                                   *
*              3770 BATCH                                           *
*                                                                   *
*********************************************************************
IBM3770  MODEENT LOGMODE=BATCH,FMPROF=X'03',TSPROF=X'03',              *
              PRIPROT=X'A3',SECPROT=X'A3',COMPROT=X'7080',             *
              APPNCOS=#BATCH
         EJECT
         TITLE 'S3270'                                                *
*********************************************************************
*                                                                   *
*              3270 SDLC                                            *
*                                                                   *
*********************************************************************
IBMS3270 MODEENT LOGMODE=S3270,FMPROF=X'02',TSPROF=X'02',              *
              PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',             *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'IBM3600'
*********************************************************************
*                                                                   *
*              3600                                                 *
*                                                                   *
*********************************************************************
IBM3600  MODEENT LOGMODE=IBM3600,FMPROF=X'04',TSPROF=X'04',            *
              PRIPROT=X'F1',SECPROT=X'F1',COMPROT=X'7000',             *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'INTRACT'
*********************************************************************
*                                                                   *
*              3650 INTERACTIVE                                     *
*                                                                   *
*********************************************************************
IBM3650I MODEENT LOGMODE=INTRACT,FMPROF=X'04',TSPROF=X'04',            *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'6000',             *
              APPNCOS=#INTER
         EJECT
         TITLE 'INTRUSER'
*********************************************************************
*                                                                   *
*              3650 INTERUSER                                       *
*                                                                   *
```

```
*********************************************************************
IBM3650U MODEENT LOGMODE=INTRUSER,FMPROF=X'04',TSPROF=X'04',          *
               PRIPROT=X'31',SECPROT=X'30',COMPROT=X'6000',           *
               APPNCOS=#INTER
         EJECT
         TITLE 'IBMS3650'
*********************************************************************
*                                                                   *
*              3650 SDLC                                            *
*                                                                   *
*********************************************************************
IBMS3650 MODEENT LOGMODE=IBMS3650,FMPROF=X'04',TSPROF=X'04',          *
               PRIPROT=X'B0',SECPROT=X'B0',COMPROT=X'4000',           *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'PIPELINE'
*********************************************************************
*                                                                   *
*              3650 PIPELINE                                        *
*                                                                   *
*********************************************************************
IBM3650P MODEENT LOGMODE=PIPELINE,FMPROF=X'04',TSPROF=X'04',          *
               PRIPROT=X'30',SECPROT=X'10',COMPROT=X'0000',           *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'SMAPPL'
*********************************************************************
*                                                                   *
*              3660                                                 *
*                                                                   *
*********************************************************************
IBM3660  MODEENT LOGMODE=SMAPPL,FMPROF=X'03',TSPROF=X'03',            *
               PRIPROT=X'A0',SECPROT=X'A0',COMPROT=X'0081',           *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'SMSNA100'
*********************************************************************
*                                                                   *
*              3660                                                 *
*                                                                   *
*********************************************************************
IBM3660A MODEENT LOGMODE=SMSNA100,FMPROF=X'00',TSPROF=X'00',          *
               PRIPROT=X'00',SECPROT=X'00',COMPROT=X'0000',           *
               APPNCOS=#CONNECT
         EJECT
         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'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'88F8',PSERVIC=X'0200000000000C280C507F00',   *
               APPNCOS=#CONNECT
         EJECT
```

```
         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'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'88F8',PSERVIC=X'02000000000185000007E00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D6327803'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3278 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'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'88F8',PSERVIC=X'02000000000185020507F00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D6327804'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3278 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'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'88F8',PSERVIC=X'02000000000018502B507F00',   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D6327805'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3278 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'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'88F8',PSERVIC=X'02000000000018501B847F00',   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D6328902'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3289 MODEL 2 PRINTER                   *
*                                                                   *
```

```
*********************************************************************
D6328902 MODEENT LOGMODE=D6328902,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'8787',PSERVIC=X'03000000000185018507F00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D6328904'
*********************************************************************
*                                                                   *
*              3276 SNA WITH 3289 MODEL 4 PRINTER
*                                                                   *
*********************************************************************
D6328904 MODEENT LOGMODE=D6328904,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'8787',PSERVIC=X'03000000000018502B507F00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32781'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3278 MODEL 1 SCREEN                           *
*              PRIMARY SCREEN 12 X 40 (480)                         *
*              ALTERNATE SCREEN 12 X 80 (960)                       *
*                                                                   *
*********************************************************************
D4A32781 MODEENT LOGMODE=D4A32781,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87C7',PSERVIC=X'0200000000000C280C507F00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32782'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3278 MODEL 2 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D4A32782 MODEENT LOGMODE=D4A32782,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87C7',PSERVIC=X'02000000000185000007E00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32783'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3278 MODEL 3 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*                                                                   *
*********************************************************************
D4A32783 MODEENT LOGMODE=D4A32783,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87C7',PSERVIC=X'02000000000185020507F00',    *
               APPNCOS=#CONNECT
```

```
         EJECT
         TITLE 'D4A32784'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)                            *
*                WITH 3278 MODEL 4 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 43 X 80 (3440)                      *
*                                                                   *
*********************************************************************
D4A32784 MODEENT LOGMODE=D4A32784,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               RUSIZES=X'87C7',PSERVIC=X'02000000000018502B507F00',   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32785'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)                            *
*                WITH 3278 MODEL 5 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 27 X 132 (3564)                     *
*                                                                   *
*********************************************************************
D4A32785 MODEENT LOGMODE=D4A32785,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               RUSIZES=X'87C7',PSERVIC=X'02000000000018501B847F00',   *
               APPNCOS=#CONNECT
         EJECT
         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'B1',SECPROT=X'90',COMPROT=X'3080',          *
               RUSIZES=X'87C7',PSERVIC=X'028000000000000000000300',   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32792'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)                            *
*                WITH 3279 MODEL 2 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D4A32792 MODEENT LOGMODE=D4A32792,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',          *
               RUSIZES=X'87C7',PSERVIC=X'028000000000185000007E00',   *
               APPNCOS=#CONNECT
         TITLE 'D4A32793'
*********************************************************************
*                                                                   *
```

```
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3279 MODEL 3 SCREEN                          *
*              PRIMARY SCREEN 24 X 80 (1920)                       *
*              ALTERNATE SCREEN 32 X 80 (2560)                     *
*                                                                  *
********************************************************************
D4A32793 MODEENT LOGMODE=D4A32793,FMPROF=X'03',TSPROF=X'03',       *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',         *
              RUSIZES=X'87C7',PSERVIC=X'028000000000185020507F00',  *
              APPNCOS=#CONNECT
         TITLE 'D329001'
********************************************************************
*                                                                  *
*              3290 TERMINAL                                       *
*              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 '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'028000000000185000007E00',                        *
              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'028000000000185020507F00',                        *
              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
         EJECT
         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)
*                                                                   *
*********************************************************************
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
         EJECT
         TITLE 'D4A32771'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3277 MODEL 1 SCREEN                           *
*                                                                   *
*********************************************************************
D4A32771 MODEENT LOGMODE=D4A32771,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87C7',PSERVIC=X'021000000000000000000100',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A32772'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3277 MODEL 2 SCREEN                           *
*                                                                   *
*********************************************************************
D4A32772 MODEENT LOGMODE=D4A32772,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87C7',PSERVIC=X'021000000000000000000200',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4A3290'
*********************************************************************
*                                                                   *
*              3274 MODEL 1A (LOCAL SNA)
*                WITH 3290 SCREEN                                   *
*              PRIMARY SCREEN SIZE 24 X 80                          *
*              ALTERNATE SCREEN SIZE UNDEFINED                      *
*                (WRITE STRUCTURED FIELD QUERY REQUIRED)            *
*
*********************************************************************
D4A3290  MODEENT LOGMODE=D4A3290,FMPROF=X'03',TSPROF=X'03',            *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'8787',PSERVIC=X'028000000000000000000300',    *
               APPNCOS=#CONNECT
         TITLE 'D4C32781'
*********************************************************************
```

```
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3278 MODEL 1 SCREEN                             *
*              PRIMARY SCREEN 12 X 40 (480)                           *
*              ALTERNATE SCREEN 12 X 80 (960)                         *
*                                                                     *
***********************************************************************
D4C32781 MODEENT LOGMODE=D4C32781,FMPROF=X'03',TSPROF=X'03',          *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
              RUSIZES=X'87F8',PSERVIC=X'0200000000000C280C507F00',    *
              APPNCOS=#CONNECT
         TITLE 'D4C32782'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3278 MODEL 2 SCREEN                             *
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              NO ALTERNATE SCREEN DEFINED                            *
*                                                                     *
***********************************************************************
D4C32782 MODEENT LOGMODE=D4C32782,FMPROF=X'03',TSPROF=X'03',          *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
              RUSIZES=X'87F8',PSERVIC=X'020000000000185000007E00',    *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C32783'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3278 MODEL 3 SCREEN                             *
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              ALTERNATE SCREEN 32 X 80 (2560)                        *
*                                                                     *
***********************************************************************
D4C32783 MODEENT LOGMODE=D4C32783,FMPROF=X'03',TSPROF=X'03',          *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
              RUSIZES=X'87F8',PSERVIC=X'020000000000185020507F00',    *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C32784'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3278 MODEL 4 SCREEN                             *
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              ALTERNATE SCREEN 43 X 80 (3440)                        *
*                                                                     *
***********************************************************************
D4C32784 MODEENT LOGMODE=D4C32784,FMPROF=X'03',TSPROF=X'03',          *
              PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
              RUSIZES=X'87F8',PSERVIC=X'02000000000018502B507F00',    *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C32785'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3278 MODEL 5 SCREEN                             *
```

```
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              ALTERNATE SCREEN 27 X 132 (3564)                       *
*                                                                     *
***********************************************************************
D4C32785 MODEENT LOGMODE=D4C32785,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',PSERVIC=X'02000000000018501B847F00',    *
               APPNCOS=#CONNECT
         EJECT
         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'028000000000000000000300',                           *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C32792'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3279 MODEL 2 SCREEN                             *
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              NO ALTERNATE SCREEN DEFINED                            *
*                                                                     *
***********************************************************************
D4C32792 MODEENT LOGMODE=D4C32792,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',PSERVIC=X'028000000000185000007E00',    *
               APPNCOS=#CONNECT
         TITLE 'D4C32793'
***********************************************************************
*                                                                     *
*              3274 MODEL 1C (REMOTE SNA)
*                WITH 3279 MODEL 3 SCREEN                             *
*              PRIMARY SCREEN 24 X 80 (1920)                          *
*              ALTERNATE SCREEN 32 X 80 (2560)                        *
*                                                                     *
***********************************************************************
D4C32793 MODEENT LOGMODE=D4C32793,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',            *
               RUSIZES=X'87F8',PSERVIC=X'028000000000185020507F00',    *
               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
         EJECT
         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
         EJECT
         TITLE 'D4C32771'
*********************************************************************
*                                                                   *
*               3274 MODEL 1C (REMOTE SNA)
*                 WITH 3277 MODEL 1 SCREEN                          *
*                                                                   *
*********************************************************************
D4C32771 MODEENT LOGMODE=D4C32771,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',PSERVIC=X'0210000000000000000000100',  *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C32772'
*********************************************************************
*                                                                   *
*               3274 MODEL 1C (REMOTE SNA)
*                 WITH 3277 MODEL 2 SCREEN                          *
*                                                                   *
*********************************************************************
D4C32772 MODEENT LOGMODE=D4C32772,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',PSERVIC=X'0210000000000000000000200',  *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4C3290'
*********************************************************************
*                                                                   *
*               3274 MODEL 1C (REMOTE SNA)
*                 WITH 3290 SCREEN                                  *
*               PRIMARY SCREEN SIZE 24 X 80                         *
*               ALTERNATE SCREEN SIZE UNDEFINED
*                 (WRITE STRUCTURED FIELD QUERY REQUIRED)
*                                                                   *
*********************************************************************
D4C3290  MODEENT LOGMODE=D4C3290,FMPROF=X'03',TSPROF=X'03',           *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',PSERVIC=X'0280000000000000000000300',  *
               APPNCOS=#CONNECT
         TITLE 'D4B32781'
*********************************************************************
```

```
*                                                                     *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)               *
*                WITH 3278 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'71',SECPROT=X'40',COMPROT=X'2000',            *
               RUSIZES=X'0000',PSERVIC=X'0040000000000C280C507F00',    *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4B32782'
***********************************************************************
*                                                                     *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)               *
*                WITH 3278 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'71',SECPROT=X'40',COMPROT=X'2000',            *
               RUSIZES=X'0000',PSERVIC=X'00400000000000185000007E00',  *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4B32783'
***********************************************************************
*                                                                     *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)               *
*                WITH 3278 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'71',SECPROT=X'40',COMPROT=X'2000',            *
               RUSIZES=X'0000',PSERVIC=X'00400000000000185020507F00',  *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4B32784'
***********************************************************************
*                                                                     *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)               *
*                WITH 3278 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'71',SECPROT=X'40',COMPROT=X'2000',            *
               RUSIZES=X'0000',PSERVIC=X'0040000000018502B507F00',     *
               APPNCOS=#CONNECT
         EJECT
```

```
         TITLE 'D4B32785'
*********************************************************************
*                                                                   *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR                  *
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)             *
*                WITH 3278 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'71',SECPROT=X'40',COMPROT=X'2000',          *
               RUSIZES=X'0000',PSERVIC=X'00400000000018501B847F00',   *
               APPNCOS=#CONNECT
         EJECT
         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
         EJECT
         TITLE 'D4B32792'
*********************************************************************
*                                                                   *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR                  *
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)             *
*                WITH 3279 MODEL 2 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              NO ALTERNATE SCREEN DEFINED                          *
*                                                                   *
*********************************************************************
D4B32792 MODEENT LOGMODE=D4B32792,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',          *
               RUSIZES=X'0000',PSERVIC=X'00C000000000185000007E00',   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'D4B32793'
*********************************************************************
*                                                                   *
*              3274 MODEL 1B/1D (LOCAL NON-SNA) OR                  *
*              3274 MODEL 1C (REMOTE BSC) OR 3276 (BSC)             *
*                WITH 3279 MODEL 3 SCREEN                           *
*              PRIMARY SCREEN 24 X 80 (1920)                        *
*              ALTERNATE SCREEN 32 X 80 (2560)                      *
*                                                                   *
*********************************************************************
D4B32793 MODEENT LOGMODE=D4B32793,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',          *
               RUSIZES=X'0000',PSERVIC=X'00C000000000185020507F00',   *
```

```
              APPNCOS=#CONNECT
         TITLE 'D4B3290'
*********************************************************************
*                                                                   *
*              3274 MODEL 1B/1D (LOCAL NON-SNA)
*                WITH 3290 SCREEN                                   *
*              PRIMARY SCREEN SIZE 24 X 80                          *
*              ALTERNATE SCREEN SIZE UNDEFINED
*                (WRITE STRUCTURED FIELD QUERY REQUIRED)
*                                                                   *
*********************************************************************
D4B3290  MODEENT LOGMODE=D4B3290,FMPROF=X'02',TSPROF=X'02',           *
              PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
              RUSIZES=X'0000',PSERVIC=X'00C00000000000000000300',    *
              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'030000000000185018507F00',   *
              APPNCOS=#CONNECT
         EJECT
         TITLE 'DSC4K'
*********************************************************************
*                                                                   *
*              PRINTER WITH 4K BUFFER
*                                                                   *
*********************************************************************
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
         EJECT
         TITLE 'LU03287B'
*********************************************************************
*                                                                   *
*              3287 PRINTER, LUTYPE 0
*                                                                   *
*********************************************************************
LU03287B MODEENT LOGMODE=LU03287B,FMPROF=X'02',TSPROF=X'02',          *
              PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
              PSERVIC=X'00200000000018501 8507F00',                  *
              APPNCOS=#CONNECT
         TITLE 'LU33286S'
*********************************************************************
*                                                                   *
*              3286 PRINTER, LUTYPE 3
*                                                                   *
*********************************************************************
LU33286S MODEENT LOGMODE=LU33286S,FMPROF=X'03',TSPROF=X'03',          *
              PRIPROT=X'B1',SECPROT=X'20',COMPROT=X'3080',           *
              PSERVIC=X'03000000000018502B507F00',                   *
              APPNCOS=#CONNECT
         EJECT
```

```
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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
         EJECT
         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'008000000000185000007E00',                   *
               APPNCOS=#CONNECT
         EJECT
         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                                *
*           SHOULD NOT BE USED FOR VSCS TERMINALS
*********************************************************************
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
         EJECT
         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                                *
*           SHOULD NOT BE USED FOR VSCS TERMINALS
*********************************************************************
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
         EJECT
         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                               *
*           SHOULD NOT BE USED FOR VSCS TERMINALS
*********************************************************************
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
         EJECT
         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
         EJECT
         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'028000000000185020507F00',                   *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'SNA5550N'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR SNA 5550 DEVICE
*                 WITH 3279 MODEL 2 SCREEN                          *
*               PRIMARY SCREEN 24 X 80 (1920)                       *
*               NO ALTERNATE SCREEN DEFINED                         *
*                                                                   *
*      PSERVIC LU0N3270 = '1'B (3270DS NOT SUPPORTED)               *
*********************************************************************
SNA5550N MODEENT LOGMODE=SNA5550N,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',PSERVIC=X'028800000000185000007E00',   *
               APPNCOS=#CONNECT
         TITLE 'NSX5550N'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 5550 DEVICE
*                 WITH 3279 MODEL 2 SCREEN                          *
*               PRIMARY SCREEN 24 X 80 (1920)                       *
*               NO ALTERNATE SCREEN DEFINED                         *
*                                                                   *
*      PSERVIC LU0N3270 = '1'B (3270DS NOT SUPPORTED)               *
*********************************************************************
NSX5550N MODEENT LOGMODE=NSX5550N,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',PSERVIC=X'008800000000185000007E00',   *
               APPNCOS=#CONNECT
         TITLE 'SNA5550Y'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR SNA 5550 DEVICE
*                 WITH 3279 MODEL 2 SCREEN                          *
*               PRIMARY SCREEN 24 X 80 (1920)                       *
*               NO ALTERNATE SCREEN DEFINED                         *
*                                                                   *
*      PSERVIC LU0N3270 = '0'B (3270DS SUPPORTED)                   *
*********************************************************************
SNA5550Y MODEENT LOGMODE=SNA5550Y,FMPROF=X'03',TSPROF=X'03',          *
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           *
               RUSIZES=X'87F8',PSERVIC=X'028000000000185000007E00',   *
               APPNCOS=#CONNECT
         TITLE 'NSX5550Y'
*********************************************************************
*        LOGMODE TABLE ENTRY FOR NON-SNA 5550 DEVICE
*                 WITH 3279 MODEL 2 SCREEN                          *
*               PRIMARY SCREEN 24 X 80 (1920)                       *
*               NO ALTERNATE SCREEN DEFINED                         *
*                                                                   *
*      PSERVIC LU0N3270 = '0'B (3270DS SUPPORTED)                   *
*********************************************************************
NSX5550Y MODEENT LOGMODE=NSX5550Y,FMPROF=X'02',TSPROF=X'02',          *
               PRIPROT=X'71',SECPROT=X'40',COMPROT=X'2000',           *
               RUSIZES=X'0000',PSERVIC=X'008000000000185000007E00',   *
               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',          *
               RUSIZES=X'0000',PSERVIC=X'000000000000000000000000',  *
               SSNDPAC=X'02',                                        *
               APPNCOS=#CONNECT
         EJECT
         TITLE 'DSILGMOD'
*********************************************************************
*                                                                   *
* DSILGMOD     DEFAULT NCCF DSILGMOD                                *
*              LOGMODE TABLE FOR BSC,LOCAL,SDLC 3275,3277,3278,3279 *
*                MODEL 2 OR 12, 24 X 80 SCREEN.                     *
*              MAY ALSO BE USED TO RUN MODELS 3, 4, 5, 2C OR 3C AS  *
*                MODEL 2.                                           *
*              ALSO FOR 3284, 3286, 3287, 3288, 3289 PRINTERS       *
*                VIA 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
         EJECT
         TITLE 'SNASVCMG'
*********************************************************************
*                                                                  *
*        LOGMODE TABLE 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'8585',ENCR=B'0000',                         *
               APPNCOS=SNASVCMG,                                     *
               PSERVIC=X'060200000000000000002300' *
*
*                               SPECIFY SYNC LEVEL = CONFIRM
         EJECT
         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 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 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
         TITLE 'ISTCOSDF'
*********************************************************************
*                                                                     *
*        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'060200000000000000000300',    *
               RUSIZES=X'8989',ENCR=B'0000',TYPE=0,                    *
               APPNCOS=#CONNECT
         TITLE 'QPCSUPP'
*********************************************************************
*                                                                     *
*        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
         TITLE 'IBMRDB'
*********************************************************************
*                                                                   *
*        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'060200000000000000102F00',   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
         MODEEND ,                   END OF DEFAULT TABLE ENTRIES
         END                       , END OF ISTINCLM
```

Session-Level USS Table (ISTINCDT) (MVS)

```
         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 INVALID'
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)
                  FAILED 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
```

Session-Level USS Table (ISTINCDT) (VM)

```
         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
UNDIAL   USSCMD   CMD=UNDIAL,FORMAT=PL1
         EJECT
VM       USSCMD   CMD=VM,REP=LOGON,FORMAT=BAL
         USSPARM  PARM=P1,REP=DATA
         USSPARM  PARM=LOGMODE
         USSPARM  PARM=APPLID,DEFAULT=VM
         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 INVALID'
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)
                  FAILED 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
```

Session-Level USS Table (ISTINCDT) (VSE)

```
         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
UNDIAL   USSCMD   CMD=UNDIAL,FORMAT=PL1
         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 INVALID'
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)
                  FAILED 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) (MVS)

```
************************************************************
* 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=APPLS,REP=VERB,VALUE=APPLS
   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=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=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
************************************************************
* DISPLAY APPLS COMMAND
************************************************************
   USSCMD  CMD=APPLS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* 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=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY CDRSCS COMMAND
************************************************************
   USSCMD  CMD=CDRSCS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY CLSTRS COMMAND
************************************************************
   USSCMD  CMD=CLSTRS,FORMAT=BAL
   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
*************************************************************
* DISPLAY GROUPS COMMAND
*************************************************************
   USSCMD  CMD=GROUPS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY LINES COMMAND
*************************************************************
   USSCMD  CMD=LINES,FORMAT=BAL
   USSPARM PARM=OWNER
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* 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
************************************************************
* DISPLAY MAJNODES COMMAND
************************************************************
   USSCMD  CMD=MAJNODES,FORMAT=BAL
************************************************************
* 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
************************************************************
* DISPLAY PENDING COMMAND
************************************************************
   USSCMD  CMD=PENDING,FORMAT=BAL
************************************************************
* 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 SESSIONS COMMAND
*****************************************************************
   USSCMD  CMD=SESSIONS,FORMAT=BAL
   USSPARM PARM=LIST,DEFAULT=COUNT
   USSPARM PARM=LU1
   USSPARM PARM=LU2
   USSPARM PARM=PLU
   USSPARM PARM=SCOPE,DEFAULT=ALL
   USSPARM PARM=SID
   USSPARM PARM=SLU
*****************************************************************
* DISPLAY STATIONS COMMAND
*****************************************************************
   USSCMD  CMD=STATIONS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
*****************************************************************
* DISPLAY STATS COMMAND
*****************************************************************
   USSCMD  CMD=STATS,FORMAT=BAL
   USSPARM PARM=NUM,DEFAULT=10
   USSPARM PARM=TYPE
*****************************************************************
* DISPLAY STORUSE COMMAND
*****************************************************************
   USSCMD  CMD=STORUSE,FORMAT=BAL
   USSPARM PARM=APPL
   USSPARM PARM=DSPNAME
   USSPARM PARM=JOBNAME
   USSPARM PARM=NUM,DEFAULT=100
   USSPARM PARM=POOL
*****************************************************************
* DISPLAY TABLE COMMAND
*****************************************************************
   USSCMD  CMD=DTABLE,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ONLY
*****************************************************************
* DISPLAY TERMS COMMAND
*****************************************************************
   USSCMD  CMD=TERMS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
*****************************************************************
* DISPLAY TGPS COMMAND
*****************************************************************
   USSCMD  CMD=TGPS,FORMAT=BAL
*****************************************************************
* DISPLAY TRACES COMMAND
*****************************************************************
   USSCMD  CMD=TRACES,FORMAT=BAL
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=TYPE,DEFAULT=VTAM
*****************************************************************
* DISPLAY TSOUSER COMMAND
*****************************************************************
   USSCMD  CMD=TSOUSER,FORMAT=BAL
*****************************************************************
* DISPLAY USERVAR COMMAND
*****************************************************************
```

```
   USSCMD  CMD=USERVAR,FORMAT=BAL
************************************************************
* DISPLAY VTAMOPTS COMMAND
************************************************************
   USSCMD  CMD=VTAMOPTS,FORMAT=BAL
   USSPARM PARM=COMP,REP=FORMAT,VALUE=COMPLETE
   USSPARM PARM=CUR,REP=FORMAT,VALUE=CURRENT
   USSPARM PARM=MOD,REP=FORMAT,VALUE=MODIFIED
   USSPARM PARM=OPT,REP=OPTION
* DEFAULT 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=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=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=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=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 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 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 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 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=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)
************************************************************
   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
```

Operation-Level USS Table (ISTINCNO) (VM, VSE)

```
************************************************************
* 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=APPLS,REP=VERB,VALUE=APPLS
   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=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=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
************************************************************
* DISPLAY APPLS COMMAND
************************************************************
   USSCMD  CMD=APPLS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* 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=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY CDRSCS COMMAND
*************************************************************
   USSCMD  CMD=CDRSCS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
*************************************************************
* DISPLAY CLSTRS COMMAND
*************************************************************
   USSCMD  CMD=CLSTRS,FORMAT=BAL
   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
*************************************************************
* DISPLAY GROUPS COMMAND
*************************************************************
   USSCMD  CMD=GROUPS,FORMAT=BAL
```

```
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY LINES COMMAND
************************************************************
   USSCMD  CMD=LINES,FORMAT=BAL
   USSPARM PARM=OWNER
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* 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
************************************************************
* DISPLAY MAJNODES COMMAND
************************************************************
   USSCMD  CMD=MAJNODES,FORMAT=BAL
************************************************************
* 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
************************************************************
* DISPLAY PENDING COMMAND
************************************************************
   USSCMD  CMD=PENDING,FORMAT=BAL
************************************************************
* 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 SESSIONS COMMAND
************************************************************
   USSCMD  CMD=SESSIONS,FORMAT=BAL
   USSPARM PARM=LIST,DEFAULT=COUNT
   USSPARM PARM=LU1
   USSPARM PARM=LU2
   USSPARM PARM=PLU
   USSPARM PARM=SCOPE,DEFAULT=ALL
   USSPARM PARM=SID
   USSPARM PARM=SLU
************************************************************
* DISPLAY STATIONS COMMAND
************************************************************
   USSCMD  CMD=STATIONS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
* DISPLAY STATS COMMAND
************************************************************
   USSCMD  CMD=STATS,FORMAT=BAL
   USSPARM PARM=NUM,DEFAULT=10
   USSPARM PARM=TYPE
************************************************************
* DISPLAY STORUSE COMMAND
************************************************************
   USSCMD  CMD=STORUSE,FORMAT=BAL
   USSPARM PARM=APPL
   USSPARM PARM=DSPNAME
   USSPARM PARM=JOBNAME
   USSPARM PARM=NUM,DEFAULT=100
   USSPARM PARM=POOL
************************************************************
* DISPLAY TABLE COMMAND
************************************************************
   USSCMD  CMD=DTABLE,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ONLY
************************************************************
* DISPLAY TERMS COMMAND
************************************************************
   USSCMD  CMD=TERMS,FORMAT=BAL
   USSPARM PARM=SCOPE,DEFAULT=ALL
************************************************************
```

```
* DISPLAY TGPS COMMAND
*****************************************************************
   USSCMD  CMD=TGPS,FORMAT=BAL
*****************************************************************
* DISPLAY TRACES COMMAND
*****************************************************************
   USSCMD  CMD=TRACES,FORMAT=BAL
   USSPARM PARM=IDTYPE,DEFAULT=RESOURCE
   USSPARM PARM=TYPE,DEFAULT=VTAM
*****************************************************************
* DISPLAY TSOUSER COMMAND
*****************************************************************
   USSCMD  CMD=TSOUSER,FORMAT=BAL
*****************************************************************
* DISPLAY USERVAR COMMAND
*****************************************************************
   USSCMD  CMD=USERVAR,FORMAT=BAL
*****************************************************************
* DISPLAY VTAMOPTS COMMAND
*****************************************************************
   USSCMD  CMD=VTAMOPTS,FORMAT=BAL
   USSPARM PARM=COMP,REP=FORMAT,VALUE=COMPLETE
   USSPARM PARM=CUR,REP=FORMAT,VALUE=CURRENT
   USSPARM PARM=MOD,REP=FORMAT,VALUE=MODIFIED
   USSPARM PARM=OPT,REP=OPTION
* DEFAULT 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=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=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=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=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 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 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 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 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=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)
*****************************************************************
   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
```

APPN Class-of-Service (COS) Definitions (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 resistanc
         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 resistanc
         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 resistanc
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 transmitte
      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 resistanc
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 resistanc
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 resistanc
*
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 resistanc
         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 RESISTANC
         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 transmitte
               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 resistanc
         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 resistanc
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 transmitte
      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 resistanc
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 resistanc
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 transmitte
      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 resistanc
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 transmitte
      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 resistanc
         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
```

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
```

Sample X.21 Call Progress Signal Retry Table (VSE)

```
         EJECT
ISTCPS10 CPSTAB  START
         CPSENTRY CPSCODE=41,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=42,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=43,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=44,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=45,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=46,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=47,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=48,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=49,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=51,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=52,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=71,CPSRCNT=5,CPSRDLY=5
         CPSENTRY CPSCODE=72,CPSRCNT=5,CPSRDLY=5
         CPSTAB  STOP
         END
```

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                                      @T1A*
*/*      % 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                                      @T1A*
*/*      % % 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                                        @T1A*
*/*      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                                        @T1A*
*/*      % 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                                        @04A*
*/*  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
```

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

```
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.

For further information on converting USS commands, see “Defining Operator Messages and Commands” in the *VTAM Network Implementation Guide*.

Glossary, Bibliography, and Index

Glossary

For definitions of the terms and abbreviations used in this book, refer to the *VTAM Glossary*.

Bibliography

VTAM V4R2 Publications

Following are descriptions of the books in the VTAM V4R2 library for the MVS/ESA, VM/ESA, and VSE/ESA operating systems. The books are arranged here according to the tasks they describe:

- Planning
- Installation, Resource Definition, and Tuning
- Customization
- Operation
- Writing Application Programs
- Diagnosis
- Quick Reference.

The complete set of unlicensed books in this section can be ordered using a single order number, SBOF-4394 for MVS, SBOF-7001 for VM, and SBOF-7002 for VSE.

Softcopy Information

IBM Networking Softcopy Collection Kit CD-ROM (SK2T-6012):

The softcopy library contains softcopy versions of the licensed and unlicensed books for VTAM V4R2 and the VTAM AnyNet Feature for V4R2 for MVS/ESA.

All of the unlicensed and licensed VTAM books described in this section are available in softcopy on this CD-ROM. These softcopy files can be read using any of the IBM BookManager READ programs. They can also be read with the IBM Library Reader program shipped on this CD.

In addition, this CD contains the Online Message Facility. The Online Message Facility is an OS/2 program that provides online access to information from *VTAM Messages and Codes* and other BookManager softcopy books. The facility helps network operators and system programmers operate and diagnose problems without interrupting those tasks.

The CD also contains softcopy of the unlicensed books of many other products.

Marketing Information

A Networking Overview and the following IBM Networking Previews are available:

- AnyNet
- VTAM.

Ask your IBM marketing representative for more information.

Planning

Planning for NetView, NCP, and VTAM (SC31-7122):

This book helps you plan for new products or for migrating to new releases of networking products. It describes product functions, explains benefits you can gain from using them in different situations, and address cross-product implications. The book contains cross-task reference information and storage estimates.

Planning for Integrated Networks (SC31-7123):

This book helps you plan for SNA (subarea and APPN) and TCP/IP networks. It includes discussion of protocol strategies, migration scenarios, processing goals, and management considerations.

Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM (SX75-0092):

This checklist identifies important tasks to consider and complete before you begin to install these product. The document can be reproduced and folded to fit easily in a pocket or folder for quick reference and easy portability.

VTAM Licensed Program Specifications (GC31-6490):

This flyer is the warranty for VTAM and includes:

- A list of new functions
- Descriptions of VTAM features
- Machine requirements
- Programming requirements.

VTAM Release Guide for MVS/ESA (GC31-6492): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Release Guide for VM/ESA (GC31-8089): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Release Guide for VSE/ESA (GC31-8090): This guide provides an overview of the new functions in VTAM V4R2 and includes:

- Advantages of new functions
- Planning considerations for new functions
- Effect of new functions on existing functions
- Changes to commands, definition statements, and messages
- Programming requirements, such as the release of NCP required.

VTAM Migration Guide for MVS/ESA (GC31-6491): This guide helps you upgrade VTAM V4R1, V3R4.2, V3R4.1, V3R4, or V3R3 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
  - Upward and downward compatibility
  - Software and hardware requirements
  - Storage requirements
  - Impacts of new functions and enhancements performed without changes to user interfaces
  - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
  - Changes to start options
  - Changes to buffer pools
  - Changes to definition statements
  - Changes to IBM-supplied default user-definable tables and modules
  - Changes to user-definable table macroinstructions
  - Changes to commands
  - Changes to messages
  - Changes to VTAM application programming interface
  - Changes to installation-wide exit routines
  - Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
  - Overview of each new function and enhancement introduced since VTAM V3R3
  - Pointers to other books in the library where implementation details can be found.

VTAM Migration Guide for VM/ESA (GC31-8071): This guide helps you upgrade VTAM V3R4.1 or V3R4 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
  - Upward and downward compatibility
  - Software and hardware requirements
  - Storage requirements
  - Impacts of new functions and enhancements performed without changes to user interfaces
  - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
  - Changes to start options
  - Changes to buffer pools
  - Changes to definition statements
  - Changes to IBM-supplied default user-definable tables and modules
  - Changes to user-definable table macroinstructions
  - Changes to commands
  - Changes to messages
  - Changes to VTAM application programming interface
  - Changes to installation-wide exit routines
  - Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
  - Overview of each new function and enhancement introduced since VTAM V3R4
  - Pointers to other books in the library where implementation details can be found.

VTAM Migration Guide for VSE/ESA (GC31-8072): This guide helps you upgrade VTAM V3R4, V3R3, or V3R2 to VTAM V4R2. It includes:

- Planning to upgrade to VTAM V4R2
  - Upward and downward compatibility
  - Software and hardware requirements
  - Storage requirements
  - Impacts of new functions and enhancements performed without changes to user interfaces
  - Changes to installation process
- Upgrading user interfaces to VTAM V4R2
  - Changes to start options
  - Changes to buffer pools
  - Changes to definition statements

  - Changes to IBM-supplied default user-definable tables and modules
  - Changes to user-definable table macroinstructions
  - Changes to commands
  - Changes to messages
  - Changes to VTAM application programming interface
  - Changes to installation-wide exit routines
  - Changes to control blocks
- Implementing optional functions and enhancements introduced in VTAM V4R2
  - Overview of each new function and enhancement introduced since VTAM V3R2
  - Pointers to other books in the library where implementation details can be found.

Estimating Storage for VTAM (SK2T-2007):

This interactive program helps you estimate the storage requirements for VTAM. The diskette includes:

- Step-by-step procedures
- Formulas used to calculate storage.

VTAM Overview for VM/ESA and VSE/ESA (GC31-8114):

This document is designed to be used with each book of the VTAM library. It helps you understand which functions are included with each package available for VTAM V4R2 for VM/ESA and VSE/ESA.

- VTAM V4R2 Client/Server
- VTAM V4R2 MultiDomain
- VTAM V4R2 InterEnterprise.

It also provides instructions for how to order a particular package.

VTAM Glossary (GC31-6558):

This glossary defines terms and abbreviations for VTAM and related products. It includes information from the IBM *Dictionary of Computing*, SC20-1699.

Installation, Resource Definition, and Tuning

VTAM Network Implementation Guide (SC31-6494):

This book presents the major concepts involved in implementing a VTAM network, and includes:

- Buffer pools, slowdown, pacing, storage considerations
- Implementation considerations
- Installation procedures
- Sample major node definitions
- Migration considerations
- Tables and filters
- TSO, VSCS, VCNS, and other programs that run with VTAM
- Tuning procedures
- VTAM start options.

Use this book in conjunction with the *VTAM Resource Definition Reference*.

VTAM Resource Definition Reference (SC31-6498):

This book describes each VTAM definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect VTAM. The information includes:

- IBM-supplied default tables (logon mode, USS and X.25)
- DTIGEN macroinstruction (VSCS start options)
- Major node definitions
- User-defined tables and filters
- VTAM start options.

If you are unfamiliar with the major concepts involved in implementing a VTAM network, use this book in conjunction with the *VTAM Network Implementation Guide*.

VTAM Resource Definition Samples (SC31-6499):

This book contains sample definitions to help you implement VTAM functions in your networks, and includes sample major node definitions.

Use this book in conjunction with the *VTAM Network Implementation Guide* and *VTAM Resource Definition Reference*.

Operation

VTAM Operation (SC31-6495):

This book serves as a reference for programmers and operators requiring detailed information about specific operator commands. The information includes:

- VTAM commands and start options
- Logon manager commands
- DISPLAY output examples (messages received)
- VSCS commands.

VTAM Operation Quick Reference (SX75-0205):

This book contains essential information about VTAM and VSCS operator commands.

VTAM Messages and Codes (SC31-6493):

This book describes messages, codes, and other information associated with VTAM messages and includes:

- Messages:
  - ELM messages for logon manager
  - IKT messages for TSO/VTAM
  - IST messages for VTAM network operators
  - ISU messages for sockets-over-SNA
  - USS messages
  - VSCS messages
- Codes and other information that display in VTAM messages:
  - Abend codes
  - Command and RU types in VTAM messages
  - Node and ID types in VTAM messages
  - Return codes for macroinstructions including ACB OPEN and CLOSE macroinstruction error fields, RTNCD-FDB2 return code combinations, and LU 6.2 RCPRI-RCSEC return codes
  - Sense codes including VTAM sense code hints, SNA sense field values for RPL-based macroinstructions, and 3270 SNA and non-SNA device sense fields
  - Status codes including resource status and session state codes
  - Wait state event codes and IDs
- Supplemental message-related information:
  - Message additions, deletions, and changes
  - Message flooding prevention
  - Message groups and subgroups
  - Message routing and suppression including descriptor codes, routing codes, and suppression levels for ELM, IKT, IST, and ISU messages
  - Message text and description formats
  - Message text of MSGLVL option messages including general information on the MSGLVL option
  - Message text of all VTAM network operator messages including variable field lengths
  - Online Message Facility.

Using IBM CommandTree/2 (SC31-7013):

IBM CommandTree/2 is a workstation product that enables an operator to construct commands and send them to a specified destination for processing. The VTAM command set library includes:

- VTAM commands
- Logon manager commands
- Help for commands and start options.
- VSCS commands

Customization

VTAM Customization (LY43-0063):

This book enables you to customize VTAM, and includes:

- Communication network management (CNM) routing table
- Installing tables and modules in VM
- Logon-interpret routine requirements
- Logon manager installation-wide exit routine for the CLU search exit
- VSCS data manipulation installation-wide exit routine
- TSO/VTAM installation-wide exit routines
- VTAM installation-wide exit routines:
  - Command verification exit (ISTCMMND)
  - Configuration services XID exit (ISTEXCCS) with description of IBM-supplied default exit
  - Directory services management exit (ISTEXCDM)
  - Generic resource resolution exit (ISTEXCGR)
  - SDDLU exit (ISTEXCSD) with description of IBM-supplied default exit
  - Session accounting exit (ISTAUCAG)
  - Session authorization exit (ISTAUCAT)
  - Session management exit (ISTEXCAA) with example
  - TPRINT processing exit (ISTRAEUE)
  - USERVAR exit (ISTEXCUV) with description of IBM-supplied default exit
  - Virtual route pacing window size calculation exit (ISTPUCWC)
  - Virtual route selection exit (ISTEXCVR).

Writing Application Programs

VTAM Programming (SC31-6496): This book describes how to use VTAM 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. The information includes:

- API concepts
  - Cryptography
  - RUs and exchanges
  - Session establishment and termination
- BIND area format
- Communication Network Management Interface
- Dictionary of VTAM macroinstructions
- OPEN or CLOSE errors
- Operating system differences
- Program Operator Coding requirements
- RAPI DSECTs and control block mappings
- RAPI global variables
- Resource-identification and access-method-support vector lists
- RPL-based macroinstructions
- RPL RTNCD,FDB2 codes
- User exit routines.

VTAM Programming for LU 6.2 (SC31-6497): This book describes the VTAM LU 6.2 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.) The information includes:

- Allocating and deallocating conversations
- APPCCMD macroinstructions and LU 6.2 DSECTs
- BIND image and response and ISTDBIND
- Conversation states
- Description and use of the following control blocks:
  - CNOS session limits control block
  - DEFINE control block
  - DISPLAY control block
  - RESTORE control block
- FMH-5 and PIP data
- LU 6.2 global variables
- Resource-identification and access-method-support vector lists
- RCPRI,RCSEC codes
- Sample program for retrieving RESTORE information
- Sample VTAM LU 6.2 application program
- Session- and conversation-level security and data encryption
- Sending and receiving data
- Sense codes for FMH-7 and UNBIND
- Summary of register usage
- Sync point services
- User exit routines.

Diagnosis

VTAM Diagnosis (LY43-0065): This book helps you identify a VTAM 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. The information includes:

- Command syntax for running traces and collecting and analyzing dumps
- VIT entries
- Procedures for collecting documentation (VTAM, VSCS, TSO)
- VTAM internal trace and VIT analysis tool
- FFST Probes
- Channel programs
- Flow diagrams
- Procedures for locating buffer pools
- VSCS dump and traces
- CPCB operation codes
- Storage and control block ID codes
- PIU discard reason codes
- Offset names and locations for VTAM buffer pools

VTAM Data Areas for MVS/ESA (LY43-0064): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Data Areas for VM/ESA (LY43-0103): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Data Areas for VSE/ESA (LY43-0104): This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM AnyNet Feature for V4R2 for MVS/ESA

VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA Gateway for OS/2 (SC31-6528):

This guide provides information to help you install, configure, use, and diagnose the sockets-over-SNA-gateway function for OS/2. This function allows socket applications running on a TCP/IP network to communicate with socket applications running on an SNA network.

VTAM AnyNet Feature for V4R2: Guide to Sockets over SNA (SC31-6526):

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.

VTAM AnyNet Feature for V4R2: Guide to SNA over TCP/IP (SC31-6527):

This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

MPTN Architecture Publications

Networking Blueprint Executive Overview (GC31-7057)

Multiprotocol Transport Networking: Technical Overview (GC31-7073)

Multiprotocol Transport Networking: Formats (GC31-7074)

APPC Application Suite Feature for V4R2 for MVS/ESA

APPC Application Suite User's Guide (SC31-6532):

This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC Application Suite for MVS/ESA. Although it's primary audience is the end user, administrators and application programmers may also find it useful.

APPC Application Suite Administration (SC31-6533):

This book contains the information that administrators need to configure the APPC Application Suite for MVS/ESA and to manage the APING, ANAME, AFTP, and A3270 servers.

APPC Application Suite Programming (SC31-6534):

This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Related Publications

VM/ESA: VMSES/E Introduction and Reference (SC24-5444)

VM/ESA: Service Guide (SC24-5527)

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Communicating Your Comments to IBM

VTAM™
Resource Definition Reference
Version 4 Release 2 for MVS/ESA, VM/ESA, and VSE/ESA

Publication No. SC31-6498-01

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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.

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:

  United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**
  - Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

VTAM™
Resource Definition Reference
Version 4 Release 2 for MVS/ESA, VM/ESA, and VSE/ESA

Publication No. SC31-6498-01

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

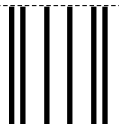
Company or Organization

Phone No.

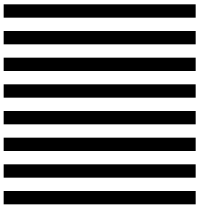
Help us help you!
SC31-6498-01

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

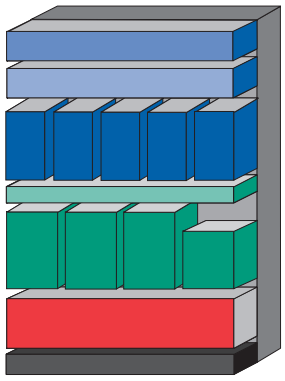
Information Development
Department E15
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

File Number: S370/4300/30XX-50
Program Number: 5695-117 (MVS/ESA)
5654-010 (VM/ESA)
5686-065 (VSE/ESA)

Printed in U.S.A.



SC31-6498-01

Spine information:

VTAM™ **Resource Definition Reference** V4R2 MVS, VM, VSE SC31-6498-01