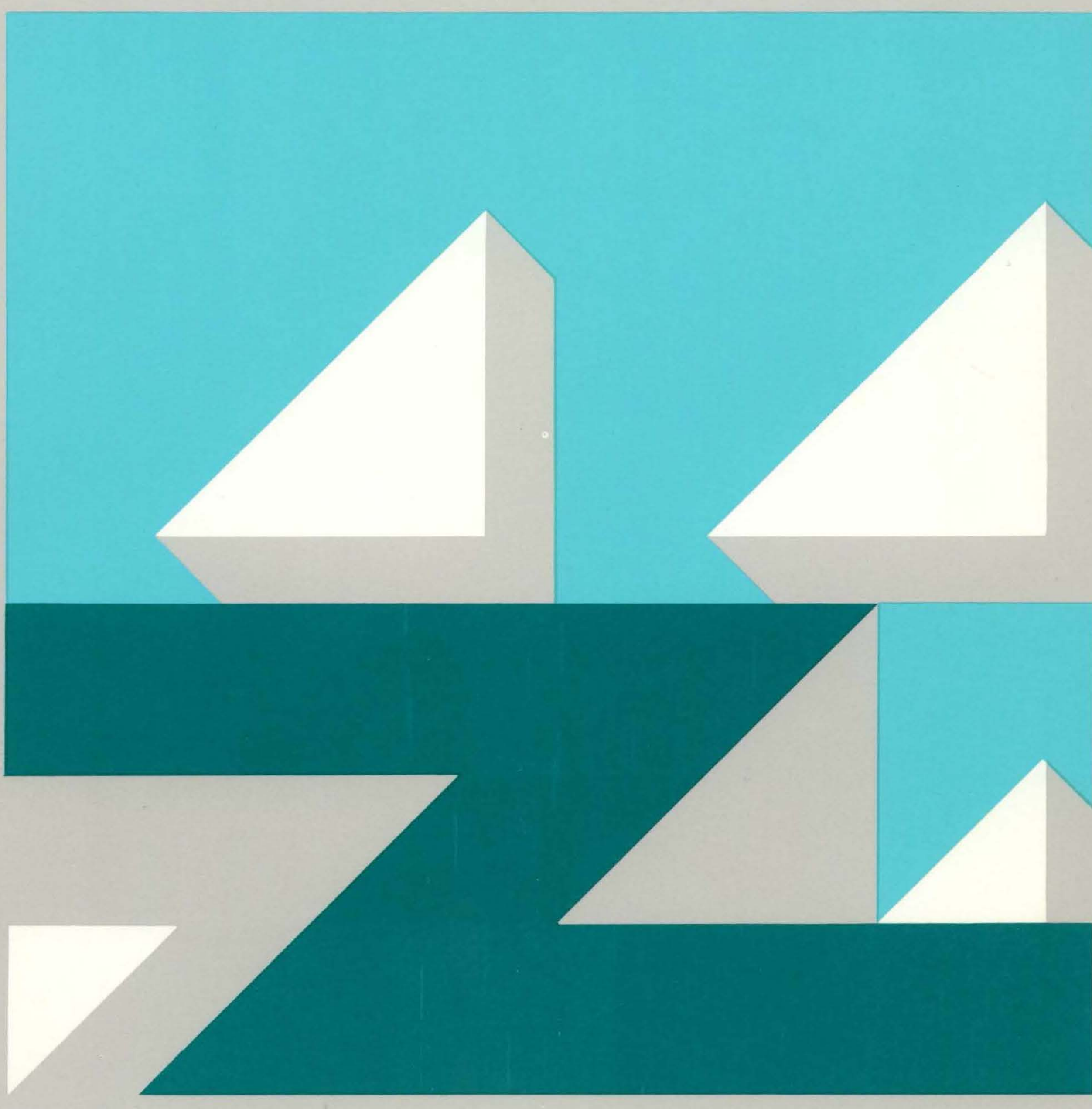
IBM

VTAM
Version 3 Releases 1 and 1.1
Installation and Resource Definition



SC23-0111-3

IBM®

VTAM
Version 3 Releases 1 and 1.1
Installation and Resource Definition

Advanced Communications Function for VTAM

Version 3 Releases 1 and 1.1 Program Numbers: 5665-313 (MVS/370), 5665-289 (MVS/XA), 5664-280 (VM), 5666-313 (VSE)

IBM.
VTAM.
003

SC23-0111-3

File No. S370/4300/30XX-50

Fourth Edition (July 1986)

This major revision obsoletes SC23-0111-2.

This edition applies to Version 3 of the Advanced Communications Function for VTAM licensed program product, operating under the:

- MVS/XA operating system (Program number 5665-289)
- MVS/370 operating system (Program number 5665-313)
- VSE operating system (Program number 5666-313)
- VM operating system (Program number 5664-280).

This edition applies to all subsequent releases and modifications until otherwise indicated in new editions.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX and 4300 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department E12, P.O. Box 12195, Research Triangle Park, North Carolina U.S.A. 27709. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1986

About This Book

This publication is for system programmers responsible for installing VTAM Version 3. It also contains information on how to define resources to VTAM.

This book contains some explanatory material, but mostly you will find *procedures* (for example, steps you should take when installing VTAM for the first time) and *reference material* (descriptions of VTAM definition statements and their operands).

If you need a broader understanding of the topics covered in this book, refer to *Network Program Products Planning.*

Who Should Use This Book

Your job title probably includes the words "system programmer", "analyst", or "administrator". Use this book if your job includes installing VTAM—transferring the VTAM object code from the product tape to your host processor—or defining the characteristics of your system after VTAM is installed.

For information about planning these activities, see *Network Program Products Planning.* For information about improving system performance after you have completed these activities, see *VTAM Customization.*

How To Use This Book

Before you begin installing VTAM and defining resources to it, you should already have planned for these tasks using *Network Program Products Planning.*

Installation means making a program ready to do useful work. It includes generating a program, installing program code into user libraries, initializing the program, and applying program temporary fixes (PTFs) to it.

Resource definition means defining the characteristics of data processing resources to VTAM. These resources include networks, hosts, Network Control Programs (NCPs), routes, terminals, and application programs.

The tasks of installation and resource definition are similar in one important aspect—both result in resources being defined. The purpose of installation is to define VTAM and its local devices and libraries to the operating system. The purpose of resource definition is to define resources to the installed VTAM program.

This book consists of the following chapters and appendixes:

Specific Categories of Information

The following reference keys are used throughout this book to indicate operating system-specific information:

MVS	Indicates information that applies to both MVS/XA and MVS/370 only.
MVS/XA	Indicates information that applies to MVS/XA only.
MVS/370	Indicates information that applies to MVS/370 only.
VM	Indicates information that applies to VM only.
VSE	Indicates information that applies to VSE only.

These keys precede the information to which they apply. Information pertaining to more than one operating system is preceded by keys for each of the operating systems that apply, as shown in the following example:

MVS VSE *This function locates the resource in this network or another network associated with a given symbolic name or network address.*

Information that applies to all the operating systems is *not* denoted by a reference key.

Where necessary, a horizontal rule is used to indicate the end of a block of information that is operating system specific and the beginning of a block of information that is common to all operating systems.

Terms Used In This Book

The following abbreviations are used in this book:

MVS	MVS, as used in this book, applies to both MVS/370 and MVS/XA unless a distinction is made.
MVS/XA	MVS/SP Version 2 for Extended Architecture.
MVS/370	MVS/SP Version 1 for System/370.
NCP	Advanced Communications Function for the Network Control Program. NCP, as used in this book, applies to both NCP and NCP subsets unless a distinction is made.
NCP Subsets	Advanced Communications Function for the Network Control Program V4 Subset.
SSP	Advanced Communications Function for System Support Programs.
TSO/VTAM	Time sharing option.
VM	VM/SP and VM/SP HPO.
VSCS	VM SNA console support.
VSE	VSE/AF.
VTAM	Advanced Communications Function for the Virtual Telecommunications Access Method.

Notes:

1. **Version** *and* **Release** *are abbreviated as* **V** *and* **R**. *For example, VTAM Version 3 Release 1.1 is abbreviated VTAM V3R1.1.*

2. *Unless otherwise indicated, the abbreviations for products refer to the latest release of the product.*

References to a *communication controller* refer to both the 3705 and 3725 communication controllers, unless indicated otherwise.

What This Book Applies To

This book applies to the Advanced Communications Function for VTAM Version 3, with Release 1 running under MVS, VM, and VSE, and Release 1.1 running under MVS and VM as described in the following table:

Full Name of Operating System	Also Called	Abbreviation	Program Number (for VTAM)
Operating System/Virtual Storage 2 Release 3.8 with Multiple Virtual Storage/System Product Version 2 for Extended Architecture	OS/VS2 Rel 3.8 with MVS/SP V2 for XA	MVS/XA	5665-289
Operating System/Virtual Storage 2 Release 3.8 with Multiple Virtual Storage/System Product Version 1 for System/370	OS/VS2 Rel 3.8 with MVS/SP V1 for 370	MVS/370	5665-313
Virtual Machine/System Product Release 4 Virtual Machine/System Product High Performance Option Releases 4 and 4.2.	VM/SP R4 VM/SP HPO R4 and 4.2.	VM	5664-280
Virtual Storage Extended/Advanced Function Version 2 Release 1 Modification Level 1	VSE/AF V2.1.1	VSE	5666-313

MVS An encryption facility is included in VTAM Version 3. It provides the encryption and decryption functions of the Encrypt/Decrypt Feature of VTAM Versions 1 and 2.

MVS/XA The extended recovery facility (XRF) is available only on MVS/XA.

VSE X.21 Short Hold Mode/Multiple Port Sharing is an addition to the VTAM Version 3 program product for VSE.

What Is New in This Book

Most of the material in this book applies to all the operating systems: MVS/XA, MVS/370, VM, and VSE. There are, however, separate sections that describe how to install VTAM in MVS, VM, and VSE.

What Is New for VTAM V3R1.1

Support for NetView

VTAM Version 3 Release 1.1 was designed to run in conjunction with the NetView program product. NetView provides network management functions, including a programmed operator and diagnostic tools.

VM VM Improvements in VTAM Version 3 Release 1.1

In VTAM Version 3 Release 1.1, VM users can take advantage of the following functions:

- SNA network interconnection
- Extended network addressing
- Support for peripheral nodes, such as the IBM 3710 Network Controller.

VM VSCS Enhancements

For VM systems, the following serviceability tools have been added for VSCS:

- VSCS dump formatter
- VSCS trace formatter.

There are also new VSCS translation exits, which you can modify, for display and keyboard devices. Two new operands on the DTIGEN macro—DEXIT and KEXIT—are used to activate these exits.

Start Options for Controlling Use of Adjacent SSCP Tables

Two start options are added to give you more control over how VTAM processes adjacent SSCP tables. They are SSCPDYN, which determines whether VTAM should add entries dynamically to the table, and SSCPORD, which determines how VTAM should scan the entries in the table.

MVS/XA Extended Recovery Facility (XRF)

XRF is a feature that you can install on VTAM Version 3 in an MVS/XA system. It provides enhanced backup capabilities for your SNA network.

One operand, HAVAIL on the APPL definition statement, is new for XRF, and one operand, LOGAPPL on the LU definition statement, has been changed.

What Is New for VTAM V3R1

VSE X.21 Short Hold Mode/Multiple Port Sharing

This is a feature that you can install on VTAM Version 3 in a VSE system. It is designed to make X.21 switched networks more flexible and more economical.

One operand, SHOLD on the GROUP and PATH definition statements, is new for this feature; the AUTODL operand on the LINE definition statement is now valid only if you code X21SW=YES on the corresponding GROUP statement.

VM VM Support

As of Version 3, VTAM now runs on the VM operating system. All functions described in this manual and all coding instructions that appear in this manual apply to VM users unless they are marked otherwise.

MVS VSE Extended Network Addressing

For MVS and VSE, VTAM's network addressing capability has been expanded in V3R1. This affects the coding of one start option and four operands. The start option is:

- HOSTSA.

The operands affected are:

- SUBAREA
- ELEMENT
- MVS ADJNETSA
- MVS ADJNETEL.

Moreover, you do not have to code the MAXSUBA start option unless VTAM Version 3 will communicate with hosts running versions of VTAM prior to Version 3, VM hosts running VTAM Version 3, or communication controllers running versions of NCP prior to Version 4.

MVS/XA 31-Bit Storage Addressing

For MVS/XA, VTAM's use of 31-bit addressing affects one start option and one operand. The start option is:

- CSALIMIT.

The operand affected is:

- MAXPVT operand of the APPL definition statement.

New Operands

Several new operands have been added for VTAM Version 3. These operands are:

- For the IBM 3710 Network Controller, SECNET is a VTAM-only operand on the PU macro instruction for NCP definition and on the PU (switched) definition for auto SSCP session restart.

- MVS VSE RECOVERY is a new operand on the CDRM definition statement.

- MVS VSE LPDATS is a new operand in dynamic reconfiguration decks.

Items That Have Been Eliminated

MVS VSE Two start options found in the previous release have been eliminated from VTAM Version 3 in MVS and VSE. They are:

- MAXAPPL
- VTAMEAS.

MVS The data set SYS1.VTAMOBJ, required in previous releases of VTAM, is obsolete.

Where To Find More Information

When using this book, you may find some of these additional publications helpful.

VTAM Publications

Figure 1 on page x shows the books in the VTAM V3R1.1 library, arranged according to related tasks.

Evaluation and Education

Network Program Products
General Information
GC23-0108

Planning

Network Program Products
Planning
SC23-0110

Installing VTAM and Defining Resources

VTAM
Installation and
Resource Definition
SC23-0111

Network Program Products
Samples
Extended Networks
SC23-0159
VM/SNA, SC30-3309

Network Program Products
Samples
Interconnected Networks
SC27-0659

Customizing VTAM

VTAM
Customization
SC23-0112

Operating VTAM

VTAM
Operation
SC23-0113

VTAM
Messages and Codes
for MVS and VSE,
SC23-0114
for VM, SC30-3275

Diagnosing VTAM Problems

VTAM
Diagnosis Guide
SC23-0116

VTAM
Diagnosis Reference
LY30-5582

VTAM
Data Areas
for MVS, LY30-5581
for VSE, LY30-5579
for VM, LY30-5580

Writing VTAM Application Programs

VTAM
Programming
SC23-0115

Reference

Network Program Products
Bibliography and
Master Index
GC23-0134

VTAM
Reference Summary
SC23-0135

Figure 1. The VTAM V3R1.1 Library

In addition to this book, the VTAM library includes:

Advanced Communications Function for VTAM Customization, SC23-0112 (referred to as *VTAM Customization*), provides system programmers with the necessary information to assist them in customizing VTAM.

Advanced Communications Function for VTAM Operation, SC23-0113 (referred to as *VTAM Operation*), serves as a reference both for network operators who operate VTAM, and for system programmers who must provide network operators with the detailed information necessary to perform this task.

Advanced Communications Function for VTAM Messages and Codes, SC23-0114 (referred to as *VTAM Messages and Codes*), contains information necessary to interpret VTAM messages and related codes. (If you are running VTAM V3R1 on a VM system, use form number SC30-3275.)

Advanced Communications Function for VTAM Programming, SC23-0115 (referred to as *VTAM Programming*), explains how to write a VTAM application program, including application program facilities and macro instructions.

Advanced Communications Function for VTAM Diagnosis Guide, SC23-0116 (referred to as *VTAM Diagnosis Guide*), shows how to identify VTAM problems and what documentation to collect before reporting them to the IBM Support Center.

Advanced Communications Function for VTAM Diagnosis Reference, LY30-5582 (referred to as *VTAM Diagnosis Reference*), provides a technical description of the VTAM function, control blocks and failure symptoms.

Advanced Communications Function for VTAM Data Areas:

- V3R1 in MVS (LY30-5581)
- V3R1.1 in MVS (LY30-5584)
- V3R1 in VSE (LY30-5579)
- V3R1 in VM (LY30-5580)
- V3R1.1 in VM (LY30-5583).

(referred to as *VTAM Data Areas*), contain tabular maps that show the contents of all control blocks and data areas used by VTAM.

Advanced Communications Function for VTAM Reference Summary, SC23-0135 (referred to as *VTAM Reference Summary*), serves as a quick reference for system programmers, application programmers, and network programmers, providing them with selected reference information that includes VTAM and VSCS commands, VTAM definition statements, VTAM start options, VTAM macro instructions, VTAM and VSCS trace formats, and selected SNA reference data.

Network Program Products Publications

You may find the following Network Program Products publications useful:

Network Program Products General Information, GC30-3350, provides an overview of a telecommunications network operating with VTAM, NCP, SSP, and NetView. It describes the hardware and software requirements, as well as the major tasks involved in using these products.

Network Program Products Planning, SC30-3351, enables system planners to select the best options in planning a network containing VTAM, NCP, SSP, and NetView.

Network Program Products Bibliography and Master Index, SC30-3353, contains a list of books that might be useful in planning, installing, or using a network containing VTAM, NCP, SSP, and NetView. It also contains a listing of topics discussed in the products' libraries.

Network Program Products Samples: Extended Networks, SC23-0159, provides tested samples of VTAM, NCP, and NetView definitions for MVS.

Network Program Products Samples: VM SNA, SC30-3309, provides tested samples of VTAM, NCP, and NetView definitions for VM.

Network Program Products Samples: Interconnected Networks, SC27-0659, describes how to implement SNA network interconnection. The samples are based on VTAM V2R2 for MVS.

Network Program Products Samples: NetView, SC30-3352, provides a guide to the samples that IBM ships with NetView. These samples are based on NetView and VTAM V3R1.1 for MVS and VM.

Network Program Products Storage Estimates, SC30-3403, provides a supplement to the *Network Program Products Planning* manual, supplying storage estimates for VTAM, NCP, CCP and NetView.

Related Products Publications

The following is a partial listing of related products publications. For a complete listing, see the *Network Program Products Bibliography and Master Index*.

Network Control Program (NCP) and System Support Programs (SSP) Publications

- *Advanced Communications Function for Network Control Program, Version 4; Advanced Communications Function for System Support Programs, Version 3: Resource Definition Guide*, SC30-3349 (referred to as *NCP-SSP Resource Definition Guide*)

- *Advanced Communications Function for Network Control Program, Version 4; Advanced Communications Function for System Support*

Programs, Version 3: Resource Definition Reference, SC30-3254 (referred to as *NCP-SSP Resource Definition Reference*)

- *Advanced Communications Function for Network Control Program, Version 4; Advanced Communications Function for System Support Programs, Version 3: Generation and Loading Guide*, SC30-3348 (referred to as *NCP-SSP Generation and Loading*)

- *Advanced Communications Function for Network Control Program, Version 4; Advanced Communications Function for System Support Programs, Version 3: Migration*, SC30-3252 (referred to as *NCP-SSP Migration*)

- *Advanced Communications Function for Network Control Program, Version 4: Customization*, LY30-5571

- *Advanced Communications Function for Network Control Program, Version 4; Advanced Communications Function for System Support Programs, Version 3: Diagnosis Guide*, LY30-5591 (referred to as *NCP-SSP Diagnosis Guide*).

NetView Publications

- *NetView Installation and Administration Guide*, SC30-3360

- *NetView Administration Reference*, SC30-3361

- *NetView Command Lists*, SC30-3362

- *NetView Customization*, LY30-5586

- *NetView Diagnosis*, LY30-5587.

Network Terminal Option (NTO) Publications

- *Network Terminal Option General Information*, GC38-0297

- *Network Terminal Option Installation*, SC38-0298.

X.25 NCP Packet Switching Interface (NPSI) Publications

- *General Information: X.21 Interface Features*, GA27-3287

- *IBM X.25 NCP Packet Switching Interface: General Information*, GC30-3189.

TCAM Publications

- *Advanced Communications Function for TCAM Version 3: General Information*, GC30-3235

- *Advanced Communications Function for TCAM Version 3: Planning Guide*, SC30-3240

- *Advanced Communications Function for TCAM Version 3: Migration*, SC30-3251
- *Advanced Communications Function for TCAM Version 3: Installation, Resource Definition, and Customization Guide*, SC30-3237
- *Advanced Communications Function for TCAM Version 3: Installation, Resource Definition, and Customization Reference*, SC30-3236
- *Advanced Communications Function for TCAM Version 3: Diagnosis Guide*, SC30-3234.

Systems Network Architecture (SNA) Publications

- *Systems Network Architecture Technical Overview*, GC30-3073
- *Systems Network Architecture Concepts and Products*, GC30-3072
- *Systems Network Architecture Sessions between Logical Units*, GC20-1868
- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic*, SC30-3112
- *Systems Network Architecture Reference Summary*, GA27-3136.

MVS/XA Publications

- *IBM System/370 Principles of Operation*, GA22-7000 (referred to as *Principles of Operation*)
- *MVS/Extended Architecture Utilities*, GC26-4018 (referred to as *Utilities*)
- *MVS/Extended Architecture Access Method Services Reference*, GC26-4019 (referred to as *Access Method Services*)
- *MVS/Extended Architecture System Programming Library: Service Aids*, GC28-1159 (referred to as *Service Aids*)
- *MVS/Extended Architecture System Programming Library: Initialization and Tuning*, GC28-1149 (referred to as *Initialization and Tuning*)
- *MVS/Extended Architecture System Generation Reference*, GC26-4009 (referred to as *System Generation Reference*)
- *MVS/Extended Architecture System Management Facilities (SMF)*, GC28-1153 (referred to as *System Management Facilities*).

MVS/370 Publications

- *IBM System/370 Principles of Operation*, GA22-7000 (referred to as *Principles of Operation*)
- *OS/VS2 MVS Utilities*, GC26-3902 (referred to as *Utilities*)
- *OS/VS2 Access Method Services*, GC26-3841 (referred to as *Access Method Services*)
- *OS/VS2 System Programming Library: Service Aids*, GC28-0674 (referred to as *Service Aids*)
- *OS/VS2 System Programming Library: Initialization and Tuning Guide (MVS/SP)*, GC28-1029 (referred to as *Initialization and Tuning*)
- *OS/VS2 System Programming Library: System Generation Reference*, GC26-3792 (referred to as *System Generation Reference*)
- *OS/VS2 System Programming Library: System Management Facilities (SMF)*, GC28-1030 (referred to as *System Management Facilities*).

VSE Publications

- *Introduction to the VSE System*, GC33-6108
- *VSE/System Package General Information*, GC33-6176
- *VSE/System Package Planning*, SC33-6177
- *VSE/System Package Installation*, SC33-6178
- *VSE/System Package System Use*, SC33-6174
- *VSE/System Package Migration*, SC33-6179
- *VSE/System Package Networking*, SC33-6180
- *VSE/System Package Diagnosis*, SC33-6182
- *VSE/Advanced Functions System Management Guide*, SC33-6094
- *VSE/Advanced Functions System Control Statements*, SC33-6095
- *VSE/Advanced Functions System Generation*, SC33-6096
- *VSE/Advanced Functions Serviceability Aids and Debugging Procedures*, SC33-6099
- *VSE/Advanced Functions System Utilities*, SC33-6100
- *VSE/Advanced Functions Maintain System History Program (MSHP) User's Guide*, SC33-6101.

VM Publications

- *VM/SP General Information*, GC20-1838
- *VM/SP Installation Guide*, SC24-5237
- *VM/SP Group Control System Guide*, SC24-5249
- *VM/SP Group Control System Macro Reference*, SC24-5250
- *VM/SP Group Control Reference Summary*, SX24-5134
- *VM/SP Planning Guide and Reference*, SC19-6201
- *VM/SP Distributed Data Processing Guide*, SC24-5241
- *Running Guest Operating Systems*, SC19-6212
- *VM/SP Terminal Reference*, GC19-6206
- *VM/SP Interactive Problem Control System Guide*, SC24-5260.

VM/HPO Publications

- *VM/SP HPO Installation Guide*, SC38-0107
- *VM/SP HPO Group Control System Guide*, ST24-5249
- *VM/SP HPO Group Control System Macro Reference*, ST24-5250
- *VM/SP HPO Planning Guide and Reference*, SC19-6223
- *VM/SP HPO Distributed Data Processing Guide*, SQ24-5241
- *VM/SP HPO Interactive Problem Control System Guide*, ST24-5260.

MVS TSO and TSO/VTAM Publications

- Each of the following books is referred to as *TSO Terminal User's Guide*. Use the one that applies to your operating system.
  - *OS/VS2 TSO Terminal User's Guide*, GC28-0645
  - *MVS/Extended Architecture TSO Terminal User's Guide*, GC28-1274.

- Each of the following books is referred to as *TSO Guide to Writing a Terminal Monitor Program*. Use the one that applies to your operating system.
  - *OS/VS2 TSO Guide to Writing a Terminal Monitor Program or a Command Processor*, GC28-0648
  - *MVS/Extended Architecture TSO Guide to Writing a Terminal Monitor Program or a Command Processor*, supplement GD23-0261 to GC28-0648
  - *MVS/Extended Architecture TSO Extensions TSO Guide to Writing a Terminal Monitor Program or a Command Processor*, SC28-1136.

- Each of the following books is referred to as *SPL: TSO*. Use the one that applies to your operating system.
  - *OS/VS2 System Programming Library: TSO*, GC28-0629
  - *MVS/Extended Architecture System Programming Library: TSO*, GC28-1173. Information, GC30-3189.

MVS Cryptographic Publications

- *IBM Cryptographic Subsystem Concepts and Facilities*, GC22-9063
- *OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility General Information*, GC28-0942
- *OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility Installation Reference Manual*, GC28-0965
- *OS/VS1 and OS/VS2 MVS Cryptographic Unit Support General Information*, GC28-1015.

Other Publications

- *IBM 3270 Information Display System: 3274 Control Unit Description and Programmer's Guide*, GA23-0061
- *IBM 3270 Information Display System: 3276 Control Unit Display Station Description and Programmer's Guide*, GA18-2081
- *IBM 3767 Models 1, 2, and 3 Communication Terminal Component Description*, GA27-3096
- *Component Description for the IBM 3776 and 3777 Communication Terminals*, GA27-3145
- *IBM 8775 Display Terminal Component Description*, GA33-3044.

Contents

Figures

Chapter 1. Introduction

An Overview of Installing VTAM

This chapter describes:

- The installation process as a whole
- Each part of the process separately
- Interdependences among the parts of the process.

In Figure 2 the installation process is broken into the following units of work:

- Installing VTAM:
  - Storing VTAM on system libraries
  - Providing support for VTAM in the operating system.
- Replacing or modifying VTAM modules. These include:
  - Installation exit routines and user replaceable modules
  - Tables for supporting session establishment and termination
  - Tables for supporting operator control.
- Writing start options, start option lists, and initial configuration lists
- Installing IBM application subsystems
- Defining the network to VTAM
- Writing NCP generation definition statements and performing an NCP generation.

The program products associated with VTAM are shown on the left of Figure 2. Their status at the end of the process is shown on the right.

This page has been intentionally left blank.

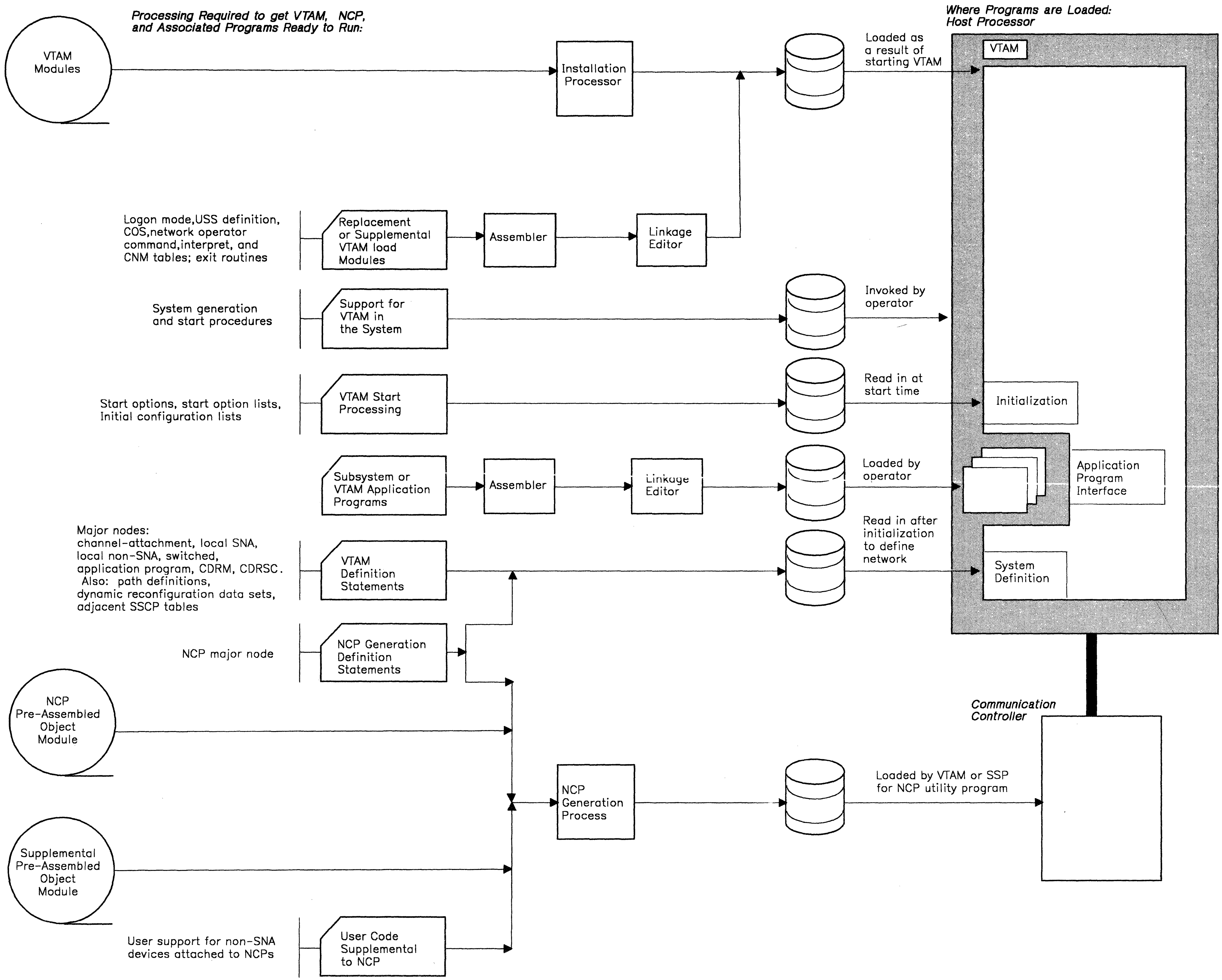


Figure 2. Overview of the Process of Installing VTAM

This page has been intentionally left blank.

In Figure 2 it is assumed that you have already installed the appropriate release of the operating system under which VTAM is to run. Installation of SNA terminal subsystems is not shown. Refer to the appropriate subsystem publication for more information.

Other things related to installing VTAM are:

- Installing network management program products, such as NetView and the Network Communication Control Facility (NCCF), and communication network management (CNM) applications to perform network management functions (see the appropriate installation manuals)

- Setting up normal, as well as backup and recovery, operating procedures (see *Network Program Products Planning* and *VTAM Operation*)

- Tuning VTAM (see *VTAM Customization*)

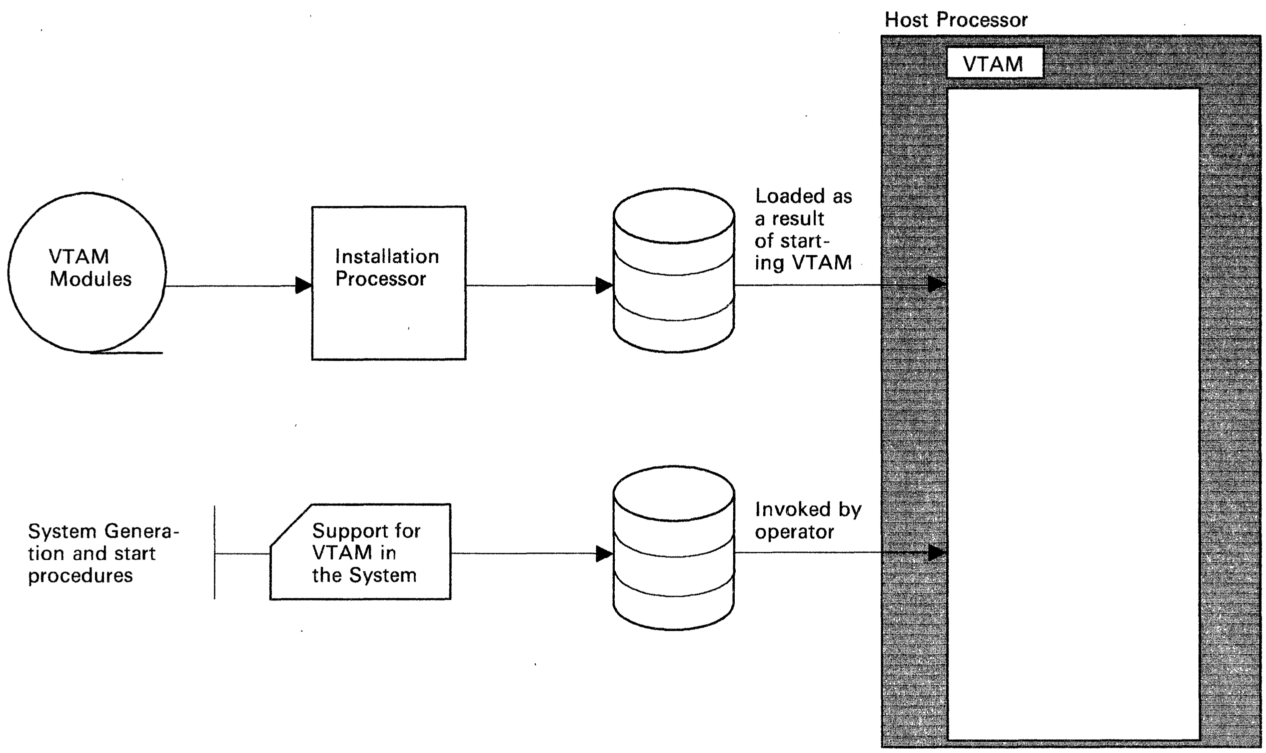
- Verifying the installation (see Chapter 8, "Verifying the Installation" on page 299).

If you are upgrading your system from a previous release level, the tasks described below will involve revising or adding to work already done, rather than performing all new work. For example, when working on VTAM definition statements, you will need to know what has changed from the original statements. In this case, you should refer to the appropriate sections in *Network Program Products Planning.*

Installing VTAM

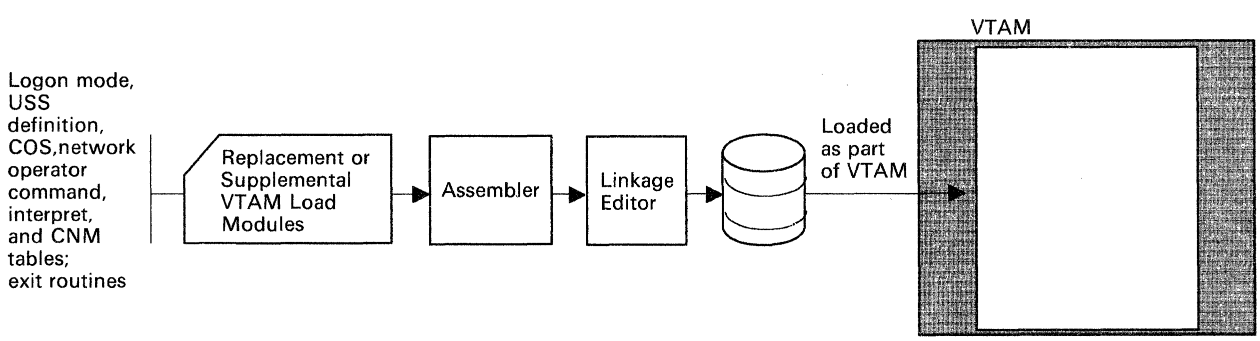
Figure 2 illustrates the process by which you combine the VTAM modules and definitions and install them on system libraries. For details of the process, refer to:

- Chapter 2, "MVS Installing VTAM in MVS" on page 13
- Chapter 3, "VM Installing VTAM in VM" on page 31
- Chapter 4, "VSE Installing VTAM in VSE" on page 53.



You must also identify VTAM to the operating system, as well as identify the local devices VTAM will support and the files and libraries it will require. This identification requires a system generation. You should also write one or more start procedures.

Replacing or Modifying VTAM Load Modules



As illustrated in the above detail from Figure 2, you can make replacements or supplements to VTAM modules after VTAM has been installed. You can add:

- Routines for supplementing the VTAM process of establishing and terminating sessions. These are the session management, virtual route selection, authorization, and accounting exit routines, and the virtual route pacing window size calculation.

- A VTAM constants module. This module would replace a module in VTAM containing constants used in regulating various functions. You can also replace selected constants.

- Tables containing information used in session establishment and termination. You can code tables to replace interpret tables in VTAM. You can also code supplementary logon mode tables for additional logon modes, and supplementary USS tables to replace IBM-supplied defaults and translate verbs and operands used by terminal operators at your installation into IBM-supplied verbs, operands, and defaults. To set up classes of service and associate them with virtual routes, you must code a class of service (COS) table.

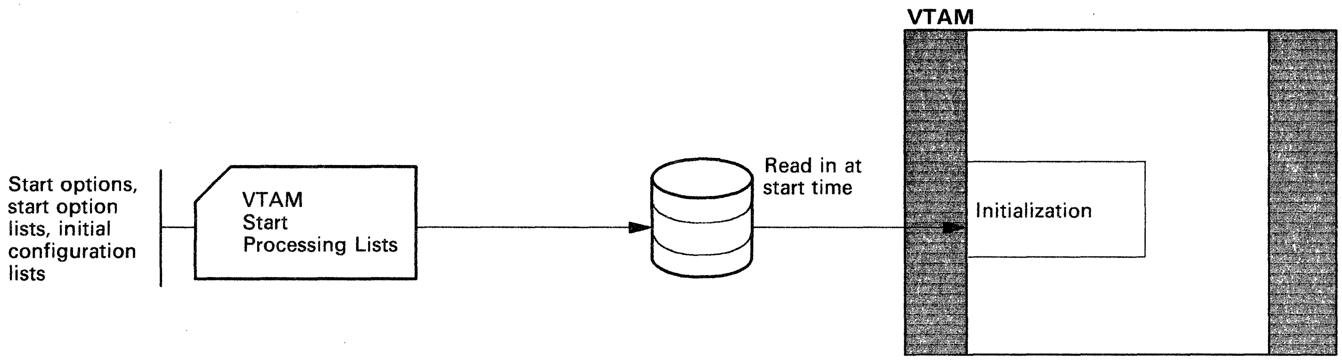
- A network operator command table containing certain VTAM operator command operands and defaults. You can code a table to supplement the one supplied by IBM. This allows you to change the defaults of operands and translate user-defined operands to IBM-supplied operands.

- A USS table for VTAM operator messages and commands. If you want to change the message text (for example, to provide non-English text) or other characteristics of a message, or to change the syntax or default values for a command, you can write a supplemental USS table.

- A supplementary communications network management (CNM) table for routing unsolicited network services request units to CNM application programs, if user-written CNM applications are used. (The IBM-supplied CNM routing table contains entries for IBM CNM products. If these are insufficient for your installation, you will have to code your own CNM routing table.)

For more information on supplementing VTAM load modules, see *VTAM Customization.*

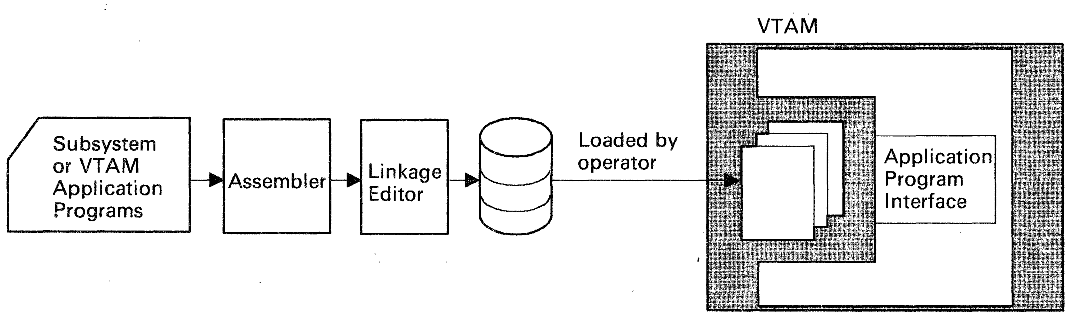
Coding Start Options, Start Option Lists, and Initial Configuration Lists



Using start option lists, initial configuration lists, and start procedures can reduce operator involvement in starting VTAM. As illustrated in the above detail from Figure 2, you can code these items and store them on libraries from which they can be retrieved at start time.

For more information on coding start options, start option lists, and initial configuration lists, see Chapter 6, "Defining Start Options" on page 217.

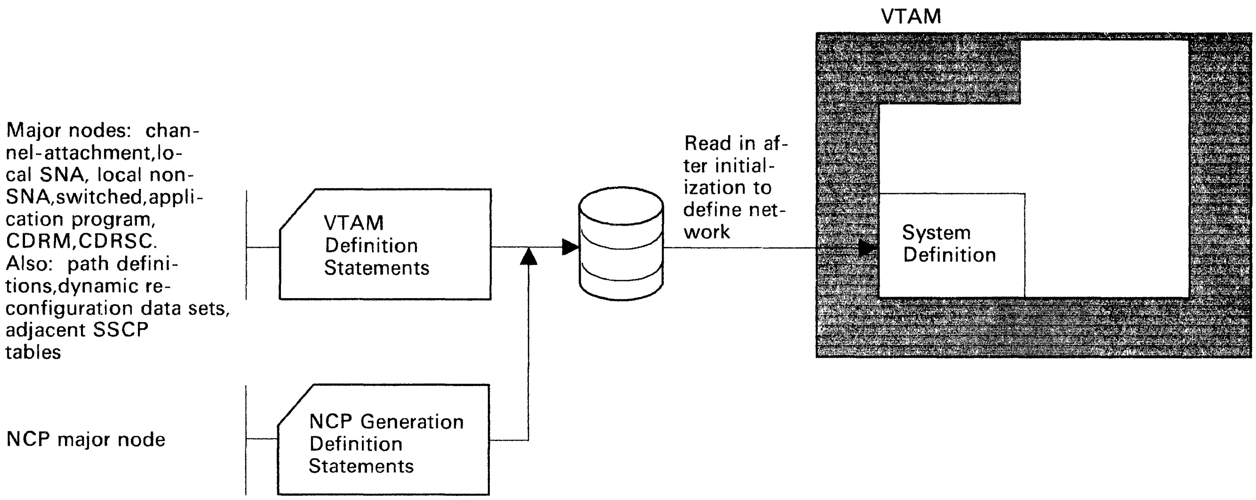
Writing VTAM Application Programs and Installing Application Subsystems



Application programs and subsystems request telecommunication services of VTAM using VTAM macro instructions. As shown in the above detail from Figure 2, you code, assemble, and link‑edit VTAM application programs, just as you would any application program. The operator initiates execution of these programs. Application programs identify themselves to VTAM before receiving telecommunication services, and the identifier they use should agree with the name assigned to them in the statements defining them to VTAM. For more information on writing VTAM application programs, see *VTAM Programming.*

There are also IBM‑written application programs described in *Network Program Products General Information.*

Defining the Network to VTAM



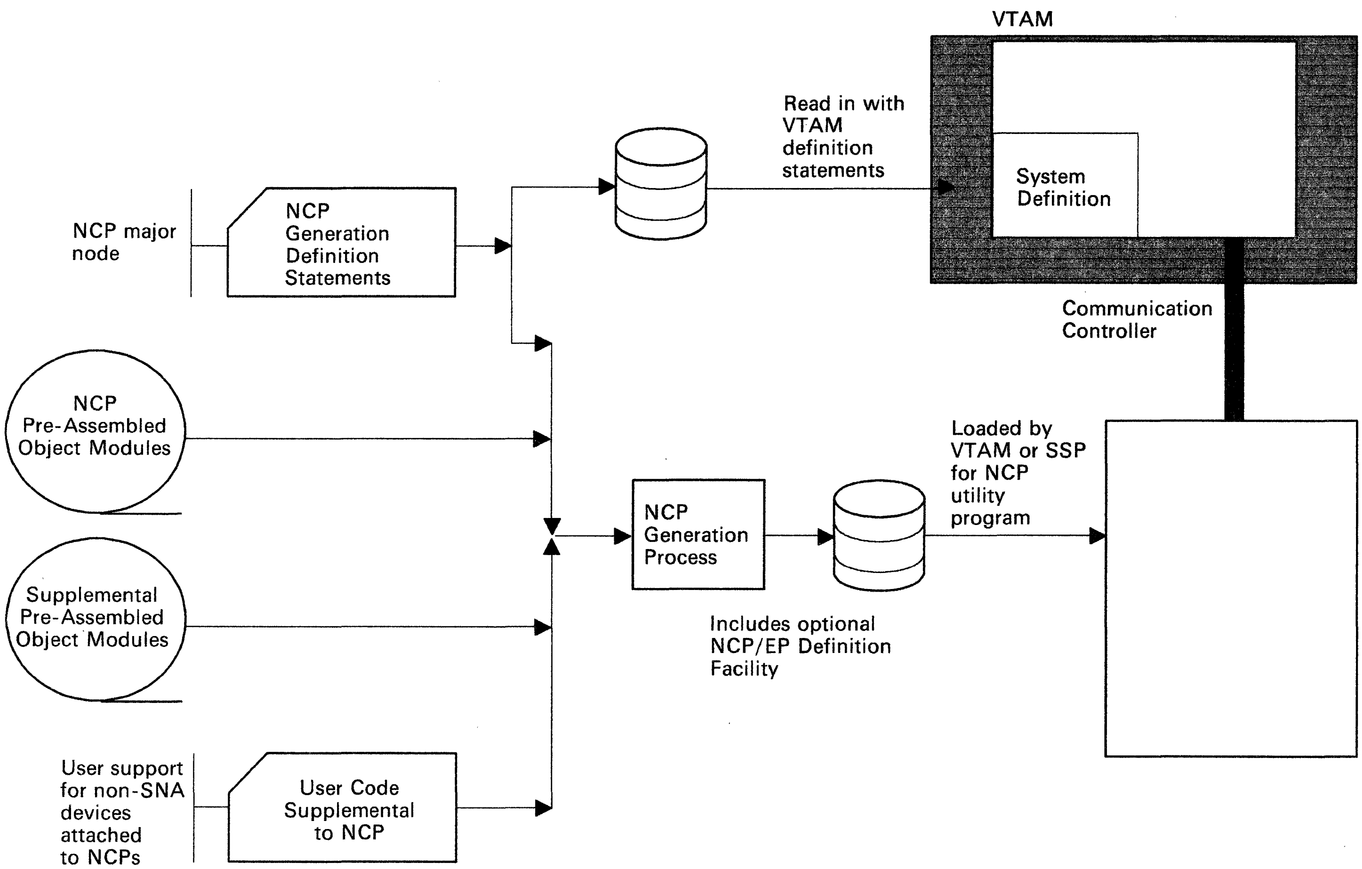
To control telecommunication, VTAM needs a definition of each resource in its domain. As illustrated in the above detail from Figure 2, you define resources that are controlled by an NCP through definition statements used to generate the NCP. Optionally, you can use the NCP/EP Definition Facility to process these NCP definition statements for improved performance. You define all the remaining resources through VTAM definition statements. For more information, see *NCP–SSP Installation and Resource Definition Guide* and *NCP–SSP Generation and Loading Guide.*

Depending on your installation requirements, you may also want to define resources outside of your domain. These include resources in other domains (see "Defining Cross–Domain Resources" on page 208 for details) and in other networks (see "Defining Cross–Domain Resources" on page 271).

Coding NCP Generation Definition Statements and Performing an NCP Generation

The network control program that is loaded into a communication controller includes the following:

- NCP modules assembled in response to generation definition statements you have written to describe the resources and functions of the NCP.
- Preassembled NCP modules for basic functions performed by an NCP.



Optionally, the network control program may also include the following:

- Modules of the IBM Network Terminal Option (NTO) program product(or a program that you have coded): To support non-SNA devices.

- Modules of the IBM Network Routing Facility program product: To allow certain devices attached to the communication controller to exchange data without routing the data through the host.

- Modules of the IBM X.25 NCP Packet Switching Interface (NPSI) program product: To allow the communication controller to be attached to an X.25-based packet-switched data network.

For a description of all the program products that can be generated along with the NCP and loaded into the communication controller, see *Network Program Products Planning.*

All these modules are link-edited and stored in libraries that must be allocated to VTAM when it is started.

After an NCP is generated, the generation definition statements are filed in the VTAM definition library, so that the NCP characteristics and attached devices will be defined to VTAM.

Post – Installation Procedures

When you complete the tasks described above, you will be ready to begin operating VTAM. However, before beginning day – to – day operations, you should:

- Set up operating procedures. You should begin to think about this task early in the installation process and continue to think about it as you verify the installation. (See *VTAM Operation.*)
- Verify the installation. Read Chapter 8, "Verifying the Installation" on page 299 for a description of this process.

After you are successfully performing day – to – day operations using VTAM, you will want to consider tuning it to perform as efficiently as possible. For more information on tuning VTAM, see *VTAM Customization.*

Interdependences

The order of the steps shown in Figure 2 is not mandatory. Ideally, they should be done in parallel. There are interdependences among the parts of the process, however, that suggest a particular sequence to follow. Some of these interdependences are, for example:

- VTAM and its supporting features can be installed in system libraries first, and replacement or supplementary modules can be added later as they are coded.
- VTAM definition statements and NCP generation definition statements have some operands that affect each other. You should be aware of these relationships, even if you do not code the definitions in parallel. These operands are described in Chapter 5, "Defining the Network to VTAM" on page 69.
- VM whenever you change VTAM definition statements or start option lists in VM, you must use the GCS ACCESS command to re – access the disk that contains the corresponding VTAMLST files before you activate the resources or restart VTAM.
- The coding of initial configuration lists, configuration restart files, and several of the start options should be coordinated with network definition for functions such as network activation, network deactivation, and backup and recovery.
- If you use the extended network addressing of VTAM Version 3, be careful to maintain address compatibility in networks that use releases of VTAM and NCP for which extended network addressing is not supported. Both the subarea and element portions of addresses must be compatible for nodes to communicate with one another.

For more information on interdependences, see *Network Program Products Planning.*

MVS VM SNA Network Interconnection Considerations

Considerations for interconnected networks are discussed in Chapter 7, "MVS VM SNA Network Interconnection Considerations" on page 247. These considerations include VTAM definition statements and operands, NCP generation definition statements, and VTAM start options.

Chapter 2. MVS Installing VTAM in MVS

Overview of the Process

VTAM is an access method of the MVS system. Before you can use VTAM, it must be identified and incorporated in the system during system generation. Once you have identified VTAM to the system, it will not be necessary to again perform a system generation for this system when upgrading to a new release of VTAM. Information on planning and generating VTAM support is given in the *System Generation Reference.*

The procedure for installing VTAM from the tape, including suggested coding statements, is described in the Memo to Users and the Program Directories that accompany the VTAM product. The Program Directory also describes the contents of the tape and space allocations required on system data sets. The tape contains the VTAM modules and macro definitions (including TSO/VTAM) needed to make VTAM part of your operating system. The System Modification Program (SMP) is used to install the product tapes. See the *OS/VS System Modification Program (SMP) System Programmers Guide* for information on installing these tapes.

For operating system requirements for VTAM, see *Network Program Products General Information.*

Figure 3 on page 14 illustrates the process of installing VTAM.

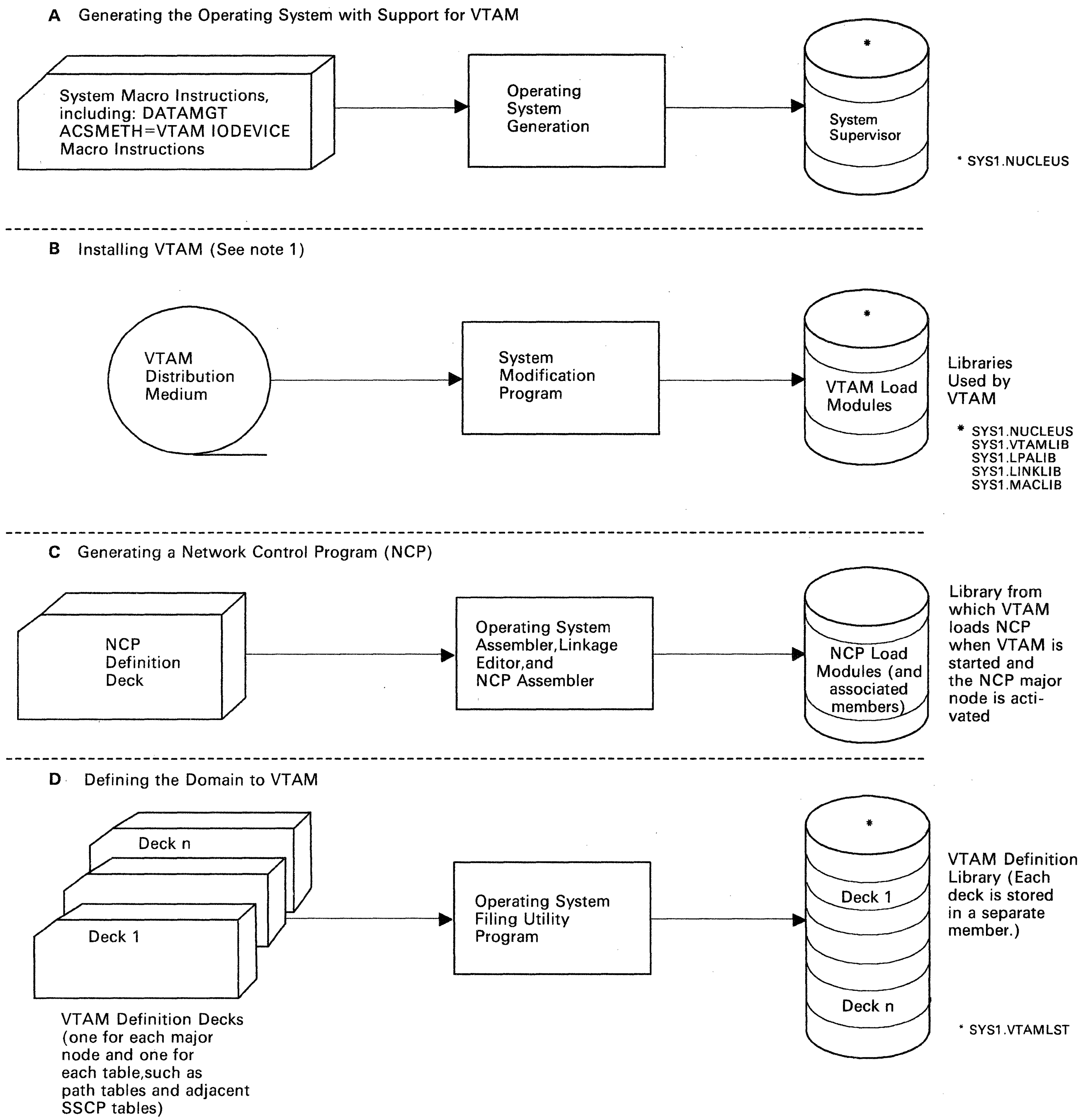
Identifying VTAM to the Operating System

If a system generation is required to include the VTAM modules and macros from the product tapes into the operating system:

- In the DATASET macro instruction (or IEHPROGM), name the system data sets that will be used by VTAM, NCP, and any SNA terminal subsystem requiring host support. System data sets are described in Figure 5 on page 21.

- In the DATAMGT macro instruction, to include VTAM in the operating system, code:

  ACSMETH=VTAM or IND=YES



Note 1: If a system generation is to be performed,VTAM can be installed on the system distribution libraries using SMP for the SYSGEN.

Figure 3 (Part 1 of 3). Creating a VTAM Domain in MVS

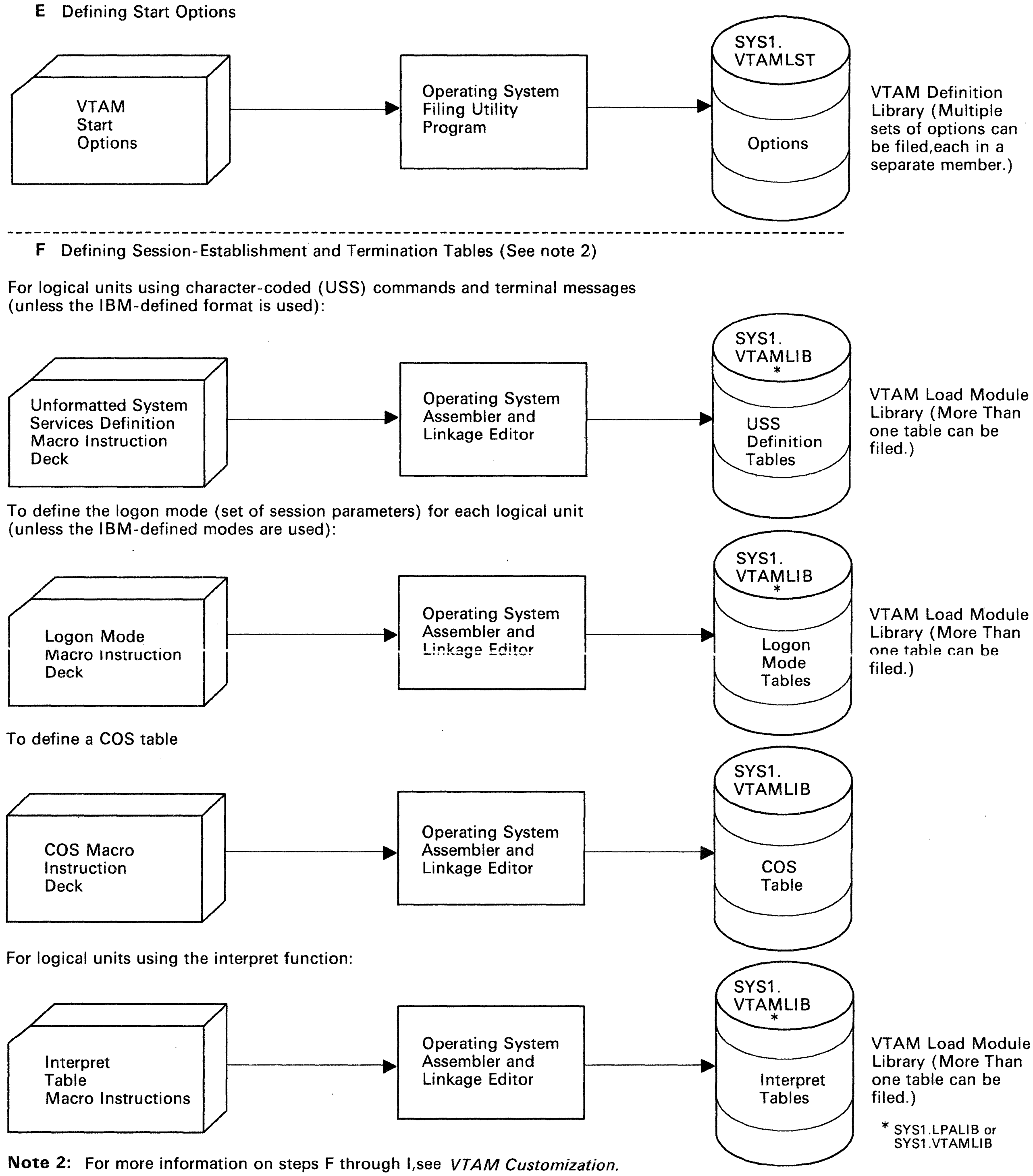
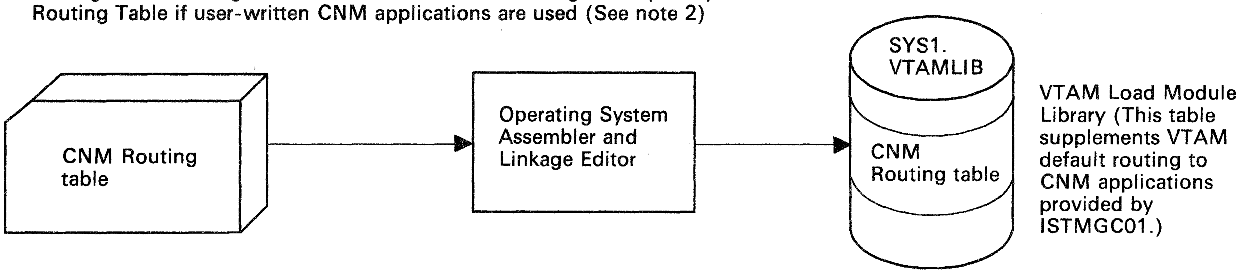
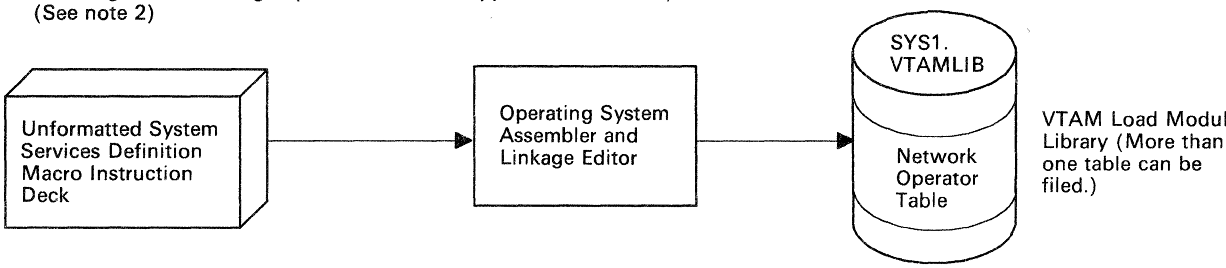


Figure 3 (Part 2 of 3). Creating a VTAM Domain in MVS

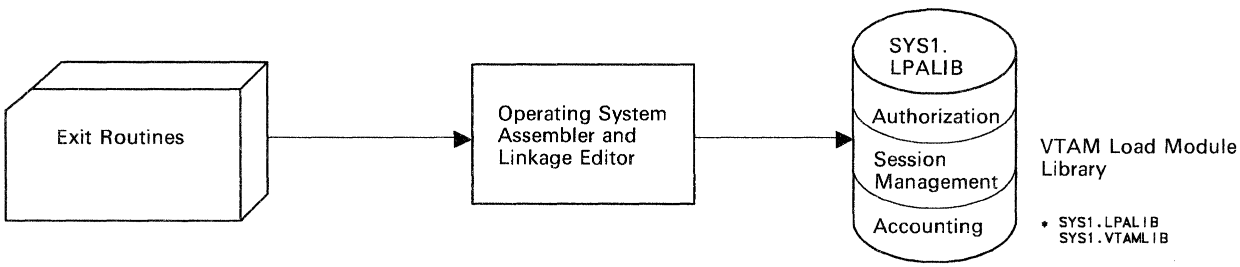
G Coding and Including a Communication Network Management (CNM) Routing Table if user-written CNM applications are used (See note 2)



H For VTAM and program operators using certain operator commands and receiving VTAM messages (unless the IBM-supplied table is used) (See note 2)



I Coding and including Exit Routines (See note 2)



Note 2: For more information, see *VTAM Customization.*

Figure 3 (Part 3 of 3). Creating a VTAM Domain in MVS

- In the SECONSLE macro instruction, if the system is generated with the multiple console support option, VTAM requires a routing code of at least ROUTCDE=(1,2,8) and command code of VALDCMD=(1,2). Depending on the message, VTAM also uses routing codes 4, 6, and 10 in combination with basic codes 1, 2, and 8.

- If the routing codes provided for VTAM messages do not meet your needs, you can change the routing codes used on the messages by coding a system user exit routine (if there is multiple console support). The exit routine receives control before messages are routed so it can examine the messages' routing codes (and descriptor codes, if desired) and change them. The system uses the modified routing codes to route these messages. To change routing codes, follow these steps:

  1. Prepare the Write To Operator/Write To Operator with Response (WTO/WTOR) exit routine and add it to the control program. The WTO/WTOR exit routine can be inserted into the resident portion

(communications task) of the control program either before or after system generation.

2. Refer to *VTAM Messages and Codes* for the message routing codes and decide on the new routing codes you would like to reassign to each message.

- If you are adding channel-attached devices either as part of system generation or as an I/O generation, the IODEVICE macro instruction should include:

  - Channel-attached communication controllers and other channel-attached devices. Devices attached to communication controllers are not defined in IODEVICE macro instructions; they are defined in statements filed for use by VTAM and the NCP. If the NCP is to be generated with partitioned emulation programming (PEP), PEP lines also must be defined with the IODEVICE macro instruction.

  - ADDRESS and OPTCHAN as the primary and secondary addresses of communication controllers, if using a 2-channel switch on communication controller channel adapters.

- MVS/XA Be sure that the extended CSA value—defined by the second value on the CSA parameter in member IEASYSxx—is at least 4 megabytes.

- In the CTRLPROG macro instruction, set CSA equal to or greater than 800.

- SYS1.VTAMLIB must be named in the APFLIB operand of the IODEVICE macro instruction when doing a system generation.

Figure 4 on page 18 shows an example of system generation statements used to include VTAM.

```
//MVSGEN JOB MSGLEVEL=(1,1),TIME=1440
// EXEC PGM=ASMBLR,REGION=1042K
//SYSLIB DD DSN=SYS1.AGENLIB,DISP=SHR,UNIT=3330,VOL=SER=DLIB01
//SYSUT1 DD UNIT=3330,SPACE=(CYL,(30,10))
//SYSUT2 DD UNIT=3330,SPACE=(CYL,(30,10))
//SYSUT3 DD UNIT=3330,SPACE=(CYL,(20,10))
//SYSPUNCH DD UNIT=3330,VOL=SER=Volser,DISP=SHR,DSN=dsname
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
*********************************************************************
*               SYSTEM CHANNELS
*********************************************************************
              CHANNEL macros go here

*********************************************************************
*               I/O DEVICES BY CHANNEL ADDRESS
*********************************************************************

              IODEVICE macros go here

*********************************************************************
*               SYSTEM OPTIONS
*********************************************************************

DMGT      DATAMGT   ACSMETH=(VTAM),IND=YES              **Note 1**

CTLPGM    CTRLPROG  CSA=800,
          APFLIB=(SYS1.VTAMLIB,volser,qual.ncplib,volser)

*********************************************************************
*               DATASETS
*********************************************************************

        DATASET  VTAMLIB,VOL=(volser,devtype),SPACE=(**Note 2**)
            (plus all other required DATASET macros)

*********************************************************************
*               CONSOLES
*********************************************************************

MSTRC     CONSOLE  MCONS=009,ALTCONS=3E1,ROUTCDE=ALL

SECCON    CONSOLE  SECONS=3E1,ALTCONS=009,ROUTCDE=ALL,VALDCMD=(1,2,3)

*********************************************************************
*               GENERATION MACRO
*********************************************************************

              The GENERATE macro goes here

        END
/*
```

Notes:

1. *If either IND=YES or ACSMETH=VTAM or both are coded, VTAM will be included in the system.*
2. *Space requirements can be computed from Network Program Products Planning or the Program Directory.*

Figure 4. Example of System Generation Statements

Allocating and Cataloging Data Sets

This section describes the data sets that must be created when installing VTAM or other related products.

The following types of data sets are used in conjunction with VTAM:

- VTAM–related data sets
- NCP–related data sets
- Trace data sets
- Cryptographic key data set.

Figure 5 summarizes the operating system data sets that contain VTAM– and NCP–related information, while Figure 6 summarizes the data sets used specifically for VTAM functions. The following discussion provides additional information about some of these data sets.

VTAM–Specific Data Sets

The following data sets contain information that is specific to VTAM.

SYS1.VTAMLIB

This is a required partitioned data set that contains VTAM load modules and user–defined tables and exit routines. This data set must be allocated and cataloged during system generation. It must be on a direct–access volume, which can be the system resident volume, and secondary space can be allocated. Space requirements are described in the Program Directory with the VTAM distribution tape.

This library can contain the interpret table (or tables), containing logon descriptions and any installation–coded logon routines that are in these tables, along with logon mode tables, the USS table, and the CNM routing table.

The DD name for SYS1.VTAMLIB must be VTAMLIB. The following DCB subparameters are recommended:

RECFM=U,BLKSIZE=7294

SYS1.VTAMLIB must be an authorized library (one of the standard names for the system library).

SYS1.VTAMLST

This is the VTAM definition library in which definition statements for each major node are filed. It is a required partitioned data set and must be allocated on a direct–access volume before filing VTAM network definitions.

The data set can be allocated and cataloged at any time before its first use by using the IEHPROGM utility program, or with the IEBUPDTE utility program when the data set is to be first used. Sufficient space should be allocated to accommodate filing of the definition decks for all types of major nodes as well as all anticipated sets of start options. Start options are filed in members under the name of ATCSTR*xx* (where *xx* represents two alphameric characters). Members named ATCCON*xx* contain lists of major nodes and adjacent SSCP tables that will be activated automatically when VTAM is started.

The DD name for SYS1.VTAMLST must be VTAMLST. The following DCB subparameters are recommended:

RECFM = FB,BLKSIZE = (any multiple of 80)

It is also recommended that you code LABEL = ,RETPD = 0 on all DD statements for SYS1.VTAMLST. If you do not, an operator awareness message requiring a reply may be generated.

Note: You can modify SYS1.VTAMLST, but you must be very careful about the relationships between VTAM and NCP definition statements. These relationships are described in "NCP Definition Statements for the NCP." For example, if you change a VTAMLST member and do not change a corresponding NCP definition statement, you could cause serious errors that are hard to diagnose.

Name of Data Set	VTAM Comments	DD Statements Required to Allocate Data Sets in Procedure for Starting VTAM	Comments
CRYPTOGRAPHIC KEY	VTAM encryption keys	No	Required if using cryptographic function. (Appendix D)
SYS1.DUMP	Records of SVC DUMP	No	Optional
SYS1.LINKLIB	VTAM initialization module, used when VTAM is started	No	Required: Created during system generation.
	NCP loader utility program	Yes	Required: Added when NCP is installed. Must be in SYS1.LINKLIB, a concatenation of SYS1.LINKLIB, or a steplib in the start procedure.
	NCP dump utility program	No	Required: Added when NCP is installed.
	NCP dump bootstrap program	No	Required: Added when NCP is installed.
	Local communication controller pre-IPL testing modules (initial test modules)	Yes	Required if controller is to be tested automatically before the loading of the NCP by VTAM: Added when the NCP is installed.
SYS1.LOGREC	VTAM error records	No	Required: Created during system generation.
SYS1.LPALIB	VTAM load modules and user-written exit routines to be loaded into the shared link pack area	No	Required: Created during system generation.
SYS1.MACLIB	VTAM definitions	Yes	Required: Created during system generation
SYS1.NUCLEUS	VTAM resident SVCs and abnormal termination modules	No	Required: Created during system generation
SYS1.PARMLIB	VTAM-related information	No	Required: Created during system generation
SYS1.SVCLIB	VTAM non-resident SVCs and ERPs for local devices	No	Optional: Created during system generation
SYS1.TRACE	GTF trace records for VTAM	No	Optional: Created during system generation

Figure 5. Operating System Data Sets Used by VTAM

Name of Data Set	VTAM Comments	DD Statements Required to Allocate Data Sets in Procedure for Starting VTAM	Comments
SYS1.ASAMPLIB	Sample of network operator command table in source – language form	Yes	Required for installation, optional thereafter Provided by IBM
SYS1.SAMPLIB	Alterable copy of sample network operator command table in source – language form	Yes	Required for installation, optional thereafter Provided by IBM
SYS1.VTAMLIB	1. Load modules for VTAM 2. User – defined tables and exit routines	Yes	VTAM load modules only are required: Created during system generation. Must be in an authorized library.
SYS1.VTAMLST	VTAM definition statements and start options.	Yes	Required: Created by user before starting VTAM. Use extreme care when modifying this data set (see note on page 20).
Configuration Restart Data Sets	VTAM status of minor nodes for each major node	Yes	Required if a warm restart is to be used: Created before starting VTAM (by user)
Initial Test Routine Data Set	NCP Diagnostic Routine	Yes	Required if a test of communication controllers is desired.
SYS1.NODELST	VTAM status of major nodes	Yes	Required if restart of all previously active major nodes is desired
NCP Load Library	NCP load modules	Yes	Each NCP stored as a separate member of library: Created when generating the NCP. Must be in an authorized library.
NCP Dump Data Set	Dump records for the NCP	Yes	Required if VTAM is requested to provide a dump of an NCP: Created by user before starting VTAM.

Figure 6. VTAM – and NCP – Specific Data Sets

Configuration Restart Data Sets (Optional)

If you choose to use the configuration restart facility of VTAM, configuration restart Virtual Storage Access Method (VSAM) data sets must be defined. The DD statement defines the configuration restart VSAM data set for the NCP major node. The *ddname* must match the *ddname* on the CONFGDS operand of the PCCU definition statement for the associated NCP or the VBUILD definition statement for the associated major node.

There are no VTAM restrictions on this data set name. For a discussion of configuration restart, see *Network Program Products Planning*. See your operating system's VSAM publication for more information on the data sets required for configuration restart. Below is an example of defining a catalog entry to allocate space for a VSAM data set to contain the configuration restart data:

```
DEFINE-
  CLUSTER(NAME(LRNCKPT) -
          VOL(PUBLIC) -
          KEYS(4 0) -
          RECORDS(200 20) -
          RECORDSIZE(24 136) -
  INDEX(NAME(LRNCKPTI.INDEX) -
          TRACKS(1))
```

You can define configuration restart data sets for any of the following major nodes:

- NCP major node
- Channel-attachment major node
- Local non-SNA major node
- Local SNA major node
- Switched major node.

For networks with more than one domain, you can define configuration restart data sets for these additional major nodes:

- Cross-domain resource major node
- Cross-domain resource manager major node.

Note: When a major node definition is changed in SYS1.VTAMLST, the associated configuration restart VSAM data set should be deleted and redefined. Do not use the WARM option when activating the new definition.

A configuration restart data set must have the following characteristics:

- The data set must be indexed. Code the INDEX operand on the DEFINE command, or allow the default value to be assumed.

- A key length of 4 bytes and an offset of 0 bytes are required. Code KEYS (4 0).

- The average record size must be 24 bytes, and the maximum record must be 136 bytes. Code RECORDSIZE (24 136).

The number of records in the file should equal the number of minor nodes defined in the major node. (For a switched major node, include each PATH statement when choosing the number of records.) Therefore, the primary allocation should be the number of minor nodes in the major node, and the secondary allocation should be about one tenth as big.

NODELST Data Set

You can define a NODELST data set to maintain a list of major nodes that are active at one time. The VTAM operator supplies the name of the data set in the NODELST start option. If you choose to use the NODELST facility, VSAM data sets must be defined. For a discussion of the NODELST facility, see *Network Program Products Planning*.

See the following example of defining a NODELST VSAM data set. The DEFINE command is used to define a catalog entry and to allocate space for an indexed cluster, as follows:

```
DEFINE-
  CLUSTER(NAME(NODLST1) -
          VOL(PUBLIC) -
          KEYS(2 0) -
          RECORDS(120 20) -
          RECORDSIZE(10 10) -
  INDEX(NAME(NODLST1I.INDEX) -
          TRACKS(1))
```

A NODELST data set must have the following characteristics:

- The data set must be indexed. Code the INDEX operand on the DEFINE command or allow the default value to be assumed.
- A key length of 2 bytes and an offset of 0 bytes are required. Code KEYS (2 0).
- The average record and the maximum record must each have a length of 10 bytes. Code RECORDSIZE (10 10).

The number of records in the file should equal the number of major node and DRDS file activations that occur from the time VTAM is started until it is halted. This includes major nodes that are reactivated. The primary allocation should be about 1.2 times the total number of major nodes and DRDS files in the network, and the secondary allocation should be about 0.2 times the total number.

All other data characteristics may be defaulted. Refer to *Access Method Services* for more information.

NCP – Specific Data Sets

The data sets described below may be needed for a communication controller:

- A load data set that contains the NCP in card – image format
- A test routine data set that contains the initial test routine (one file for each host VTAM)
- A dump data set to receive the NCP dump output (one data set for each host VTAM)

- A dump data set for communication scanner processor (CSP) and for maintenance and operator subsystem (MOSS) traces.

See NCP publications for additional information about these data sets.

NCP Load Library

To load the NCP, an NCP load data set must be created (space must be allocated) and, optionally, cataloged. The name of this data set used in the JCL is defined when the NCP is coded.

The DD statement defines the data set that holds the generated NCP and RRT modules. In addition:

- The *ddname* must be the same as the unqualified name of the data set.

- The data set name (DSN operand) must match the LOADLIB operand for the BUILD definition statements for the associated NCP.

- If the QUALIFY operand is coded in the BUILD definition statement, use the qualified data set name when coding DSN. (For VTAM, the LOADLIB operand is used instead of the QUALIFY operand.)

- NCP load data sets must be in an authorized library.

A single NCP load library can contain more than one NCP.

Initial Test Routine Data Set

For a channel-attached NCP, the VTAM user can load a diagnostic routine, called the initial test routine, into a communication controller and have it execute before the NCP is loaded.

The INITEST DD statement defines the data set that holds this initial test routine for the channel-attached communication controller. This statement must be coded if INITEST=YES is coded in the PCCU definition statement of the channel-attached NCP. Normally, NCP load modules for initial test are placed on SYS1.LINKLIB by the user during NCP installation. If the initial test routine has not been placed on SYS1.LINKLIB, check the documentation accompanying the NCP installation tapes for the name of the data set initially containing the routine.

NCP Dump Data Set

To dump the NCP, you must allocate space for an NCP dump data set. You may also catalog the dump data set. Its name is defined when the NCP is coded.

This dump data set must accommodate a dump of the entire communication controller storage. The size of communication controller storage depends on the model number.

The amount of direct access storage required for the dump data set can be determined from the storage size of the communication controller and from calculating the number of tracks needed to hold the required number of 512−byte records of communication controller storage.

The DD statement defines the dump data set for the communication controller. The *ddname* must match the *ddname* in the DUMPDS operand of the PCCU definition statement for the associated NCP. There are no VTAM restrictions on the data set name.

This procedure for determining the amount of storage needed for the dump data set applies to both the IBM 3725 Communication Controller and the 3705 Communication Controller. However, the 3725 requires an additional 2K bytes.

VTAM dump processing fails if the SSP modules that are loaded to process the dump are not accessible to VTAM. For these modules to be accessible to VTAM, the data set that contains them should be a link list data set named in the LNKLSTxx member (where *xx* represents the specific parameter) of SYS1.PARMLIB. (See Figure 8 on page 29 for a sample STEPLIB DD statement.)

CSP and MOSS Dump Data Sets (NCP in a 3725 Communication Controller Only)

To dump the communication scanner processor (CSP) and maintenance and operator subsystem (MOSS) microcode for problem determination, you must create one data set for dumping each component. Optionally, these data sets can be cataloged. The names of these data sets are defined to VTAM in the start procedure. Figure 8 on page 29 shows an example of this procedure.

You can determine the amount of direct access storage required for these dump data sets from the sizes of the respective components and by calculating the number of tracks needed to hold the required number of records.

The DD statement for each dump data set defines it for the NCP utility used to dump the communication controller. The *ddname* must match the name in the CDUMPDS (for a CSP dump) or MDUMPDS (for a MOSS dump) operand of the PCCU definition statement for the appropriate NCP. VTAM has no restrictions on the data set name.

VTAM dump processing fails if the SSP modules that are loaded to process the dump are not accessible to VTAM. For these modules to be accessible to VTAM, the data set that contains them should be a link list data set named in the LNKLSTxx member (where *xx* represents the specific parameter) of SYS1.PARMLIB.

Changing NCP Load Program Attributes

A Technique for Reducing Loading Time

To reduce the time needed to load NCPs into communication controllers (especially into link – attached communication controllers), you can change the attributes for an NCP load program. The VTAM load program interface ISTINC05 loads NCPs sequentially into communication controllers. By using the linkage editor, you can change the attribute of these modules from "serially reusable" to "not reusable". This attribute causes separate copies of these NCP load programs to be brought into virtual storage for each NCP that is to be loaded, reducing loading time by providing concurrent NCP loading.

Note: The change to the NCP load module could be reset by System Modification Program (SMP) maintenance without notification. To avoid this, you could apply the NCP load module modification with SMP. For more information on using SMP, see *OS/VS System Modifications Program (SMP) System Programmer's Guide.*

Example of Changing NCP Load Program Attributes

In this example, the linkage editor program changes the load module attribute for the NCP load program (IFLOADRN) and the VTAM load program interface from "serially reusable" to "not reusable".

```
//CHGNCPAT  JOB   07550,SMITH
//LKED      EXEC  PGM=HEWL,PARM='XREF,LET,LIST,NCAL'
//SYSUT1    DD    UNIT=SYSDA,SPACE=(TRK,(100,10))
//SYSPRINT  DD    SYSOUT=A
//SYSLMOD   DD    DSN=SYS1.LINKLIB,DISP=OLD
//* MVS SYSLIN CONTROL STATEMENTS
//SYSLIN    DD    *
   INCLUDE SYSLMOD(ISTINC05)
   NAME ISTINC05(R)
/*
```

Figure 7. Example of Changing NCP Load Program Attributes

The EXEC statement determines:

- The program name (PGM = HEWL).

- That the output modules are to be marked "not reusable" (the default value) since neither the REUS nor RENT options of PARM are coded.

- That the linkage editor is to mark the output module executable (LET option) and to produce a cross – reference table of the output on the diagnostic output data set (XREF option).

- That all the control statements processed by the linkage editor are listed in card – image format on the diagnostic output data set (LIST option).

- That the linkage editor is not to call library members to resolve external references (NCAL option).

The SYSUT1 DD statement defines the space allocation for the intermediate data set.

The SYSPRINT DD statement defines the diagnostic output data set (a printer is assumed).

The SYSLMOD DD statement defines the output module library SYS1.LINKLIB (assumed to be cataloged).

The SYSLIN DD statement indicates that input in the form of linkage editor control statements follows.

The INCLUDE control statements indicate that module ISTINC05 (on SYS1.LINKLIB) is additional input to the linkage editor.

The NAME control statements indicate (1) the names for the load modules (created by the linkage editor processing), and (2) that these load modules replace identically–named modules in the output module library SYS1.LINKLIB.

Coding a VTAM Start Procedure

Recommendations for Coding a VTAM Start Procedure

You should code a VTAM start procedure and file it in SYS1.PROCLIB. The system operator specifies the procedure when starting VTAM.

In this discussion, the start procedure is called NET. This name is not required, but for consistency in entering the VTAM operator commands and to reduce the operator's chances of making a syntax error, it is recommended. NET *is* required as the first operand on the DISPLAY, HALT, and VARY network operator commands.

The first operand of the START and MODIFY network operator commands must be the name of this start procedure (NET in this example).

Example of Coding a VTAM Start Procedure

Figure 8 on page 29 lists the job control statements of a typical start procedure.

```
//NET          PROC
//VTAM         EXEC    PGM=ISTINM01,TIME=1440,REGION=4096K
//STEPLIB      DD      DSN=SYS1.SSPLIB,DISP=SHR
//VTAMLST      DD      DSN=SYS1.VTAMLST,DISP=SHR,LABEL=RETPD=0
//VTAMLIB      DD      DSN=SYS1.VTAMLIB,DISP=SHR
//NCPDUMP      DD      DSN=SYS1.NCPDUMP,DISP=SHR
//* DATA SETS FOR 3725 DUMPS
//CSPDUMP      DD      DSN=SYS1.CDUMP,DISP=SHR
//MOSSDUMP     DD      DSN=SYS1.MDUMP,DISP=SHR
//SYSABEND     DD      SYSOUT=A
//NCPLOAD      DD      DSN=SYS1.NCPLOAD,DISP=SHR
//* NODELST DATA SET
//NODEDS1      DD      DSN=NODEDS1,DISP=OLD,AMP=AMORG
//* ALTERNATE NODELST DATA SET
//NODEDS2      DD      DSN=NODEDS2,DISP=OLD,AMP=AMORG
```

Note: On the EXEC statement, ISTINM01 is a VTAM module name. You must code PGM=ISTINM01.

Figure 8. Example of Statements to Define a VTAM Start Procedure

The sample start procedure was written with these assumptions:

- A communication controller is in the network.
- The generated NCP and RRT modules for the communication controller reside on data set SYS1.NCPLOAD.
- The SSP modules needed to load and dump the communication controllers are in SYS1.SSPLIB.
- A dump data set is desired for each communication controller.
- A dump data set is desired for the communication scanner processor (CSP).
- A dump data set is desired for the maintenance and operator subsystem (MOSS).
- Two NODELST data sets have been defined and may be used by VTAM.
- Space has been allocated for all data sets and they are cataloged.

The EXEC statement causes the VTAM host attachment module to be executed. This module is ISTINM01.

Verifying the Installation

When VTAM has been installed, you can begin verifying it. See Chapter 8, "Verifying the Installation" on page 299 for suggested steps.

Chapter 3. VM Installing VTAM in VM

VTAM is an access method of the VM system. Before using VTAM, you must prepare your VM system to support it. You must:

- Generate the VM Control Program (CP), defining shared segments for VTAM and the Group Control System (GCS)
- Generate GCS, which provides supervisor and data management services for the VTAM virtual machine and other virtual machines in which VTAM application programs run
- Build the VTAM shared segment and the VTAM and VSCS load module libraries
- Define the VTAM virtual machine and other virtual machines associated with VTAM.

You will find information on planning and generating support for VTAM in:

- *VM/SP Planning Guide and Reference*
- *Group Control System Guide.*

The process for installing VTAM is shown in Figure 9 on page 32. You will use a single userid to install VTAM and to maintain it once it is in use. This userid can have any name, but MAINT is recommended. You will use another userid, VTAM, for day-to-day operation.

After you follow the setup procedures summarized in Figure 9 on page 32, the installation is done automatically by EXECs.

Refer to the Memo to Users for installation procedures and to the Program Directory that accompanies your VTAM shipment for special instructions.

1	Determine how storage is to be allocated in the VTAM virtual machine. Refer to "Determining Storage Needs for the VTAM Virtual Machine."
2	Generate CP. Create entries in DMKSNT, the CP system name table, to define discontiguous shared segments for VTAM and GCS. Create entries in DMKRIO, the CP system definition file, to define locally–attached control units. Refer to "Generating the Control Program (CP)" on page 34. (Refer to the *VM Installation Guide* for information on installing CP and CMS.)
3	Log on to the MAINT userid and IPL CMS.
4	Generate GCS. Refer to "Generating the Group Control System (GCS)" on page 35. You must generate GCS with the VTAM virtual machine authorized. VSCS must also be authorized if it runs in a separate virtual machine from VTAM. (Refer to the *Group Control System Guide* for information on setting up authorized userids.)
5	Create or update directories for the userids that VTAM and VSCS will use. Refer to: • "The Minidisks You Will Need" on page 36 • "Setting up Directory Entries for VTAM and MAINT" on page 35.
6	Install VTAM by invoking the INSTFPP EXEC. Refer to "The Installation Procedure" on page 43.
7	Verify that VTAM was installed correctly. Refer to "Verifying the Installation" on page 45.
8	Set up VTAM to start automatically at IPL time. Refer to "Coding the PROFILEs for the Virtual Machines" on page 46.

Figure 9. Summary of Installing VTAM and VSCS on a VM System

Preparing to Install VTAM

You can only run VTAM if the Conversational Monitor System (CMS) and the Group Control System (GCS) are already installed.

Determining Storage Needs for the VTAM Virtual Machine

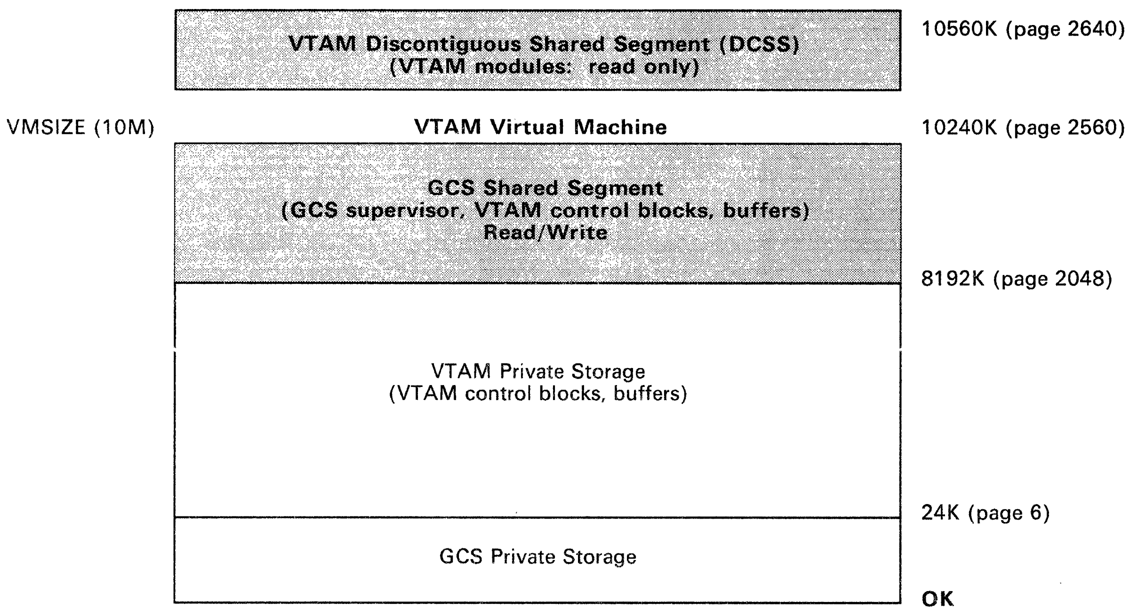
You need to decide two things: how to allocate storage in the VTAM virtual machine, and how much storage you will need.

How Storage Is Allocated

Figure 10 suggests a way to allocate storage in the VTAM virtual machine. In this example the size of the VTAM virtual machine is 10 megabytes. As you decide how to allocate storage in the VTAM virtual machine, consider:

- To avoid wasting storage, VTAM's private storage area should consist of a single block. For example, in Figure 10, this entire area is located between GCS private storage and the GCS shared segment.

- The VTAM shared segment should be higher than the highest address of the VTAM virtual machine, and higher than the highest address of any other virtual machines that attach it. (These virtual machines would include NCCF, RSCS, and the GCS recovery machine.)



VTAM NAMESYS

```
SYSNAME  = VTAM
SYSVOL   = VMSRES,
SYSSTRT  = (015, 01),
SYSPGNM  = (2560-2655)
SYSPGCT  = 96,
SYSHRSG  = (160,161,162,163,164,165)
SYSSIZE  = 2048K,
SYSCYL   = ,
VSYSRES  = ,
VYSSADR  = IGNORE,
PROTECT  = ON
```

GCS NAMESYS

```
SYSNAME  = GCS
SYSVOL   = VMGCS1,
SYSSTRT  = (01,001) ,
SYSPGNM  = (0-5, 2048-2559) ,
SYSPGCT  = 518,
SYSHRSG  = (128,129,130,131,132,133,134,
           135,136,137,138,139,140,141,
           142,143,144,145,146,147,148,
           149,150,151,152,153,154,155,
           156,157,158,159) ,
SYSSIZE  = 256K,
VSYSADR  = 595,
SYSCYL   = 021,
VSYSRES  = VMGCS1,
VMGROUP  = YES,
PROTECT  = OFF
```

Figure 10. Sample Storage Allocation for VTAM and GCS

Size of the VTAM Virtual Machine

To calculate how much storage VTAM and GCS will need, follow this procedure:

1. Determine private storage requirements for GCS. The recommended value is 24K bytes. (See the *GCS Guide* for details.)

2. Calculate private storage requirements for VTAM, using the formulas in *Network Program Products Storage Estimates*. These requirements consist of:

   - Private constant storage
   - Private dynamic storage
   - VSCS private storage (if you intend to run VSCS in the VTAM virtual machine).

3. Calculate common storage requirements for GCS. These consist of:

   - The GCS supervisor (about 252K)
   - GCS control blocks
   - The GCS trace table
   - Application – specific requirements; for VTAM, these are:
     - VTAM constant common storage
     - VTAM dynamic common storage
     - VTAM buffer pools
     - The VTAM internal trace table (size depends on the number of 4K pages allocated by the VTAM TRACE start option or operator command)
     - Storage for activating channel – attached devices.

   The GCS requirements are described in the *GCS Guide*. The VTAM requirements are described in *Network Program Products Storage Estimates*.

4. Determine common storage requirements for VTAM (the size of the VTAM shared segment). This is fixed at 288K; to allocate storage in 64K segments, you should round this to 320K.

5. Add the results from Steps 1, 2, and 3 to find the size of the VTAM virtual machine. Do not add the result from Step 4; the VTAM shared segment should be outside the VTAM virtual machine.

Generating the Control Program (CP)

When you generate CP:

- Update DMKSNT, the CP system name table, so that a discontiguous shared segment (DCSS) is defined for VTAM. The DCSS contains VTAM modules that are shared between virtual machines in the group. Its size was calculated in Step 4 on page 34. Figure 10 on page 33 shows a sample DMKSNT entry for the VTAM discontiguous shared segment.

- Update DMKSNT so a shared segment is defined for GCS. The shared segment should be large enough to contain the VTAM control blocks and buffers. Figure 10 shows a sample DMKSNT entry for the GCS shared segment.

- Update DMKRIO, the CP system definition file, to define all locally-attached communication controllers and cluster controllers that VTAM will use.

- Do not use the small CP option.

Refer to the *VM Installation Guide* for detailed information on how to generate CP.

Generating the Group Control System (GCS)

Use the GCS GROUP EXEC to generate GCS. When you generate GCS:

- Define the VTAM virtual machine as an authorized machine.

- Define the VSCS virtual machine as an authorized machine, if VSCS operates in a separate virtual machine from VTAM.

- Define an authorized virtual machine to receive dumps.

- Specify the size of the GCS trace table (the default is 16K).

- Define a VM recovery virtual machine for the VM group. This machine must be different from the VTAM virtual machine.

- Specify the maximum number of virtual machines that can exist in the system.

- Supply the name of the VTAM shared segment.

Refer to the *Group Control System Guide* for detailed information on how to generate GCS.

Setting up Directory Entries for VTAM and MAINT

The sample directory entries in Figure 12 on page 36 and Figure 11 on page 36 will help you build directories for the VTAM userid and the MAINT userid. They are samples; both show space allocated from a 3350 volume called VMVTAM. You should modify the volume ID, device type, cylinder counts, and access passwords for your own needs.

The DIAG98 option in Figure 12 is strongly recommended for the VTAM userid, unless you are using the VM/SP High Performance Option (HPO). For HPO considerations, see "Real I/O in HPO Systems" on page 38. It allows the VTAM virtual machine to fix (or "lock") pages in CP storage and execute channel programs that contain CP storage addresses. These addresses do not require translation, which improves VTAM's performance.

For complete information on setting up directories, see the *VM/SP Planning Guide and Reference.*

```
USER MAINT CPCMS 12M 16M ABCDEFG
  OPTION ECMODE DIAG98
  IUCV *CCS P M 10
  IUCV ANY P M 0
  :
  :
  MDISK 298 3350 101 003 MAINT WR RMAINT WMAINT MMAINT
  MDISK 299 3350 104 020 MAINT WR RMAINT WMAINT MMAINT
  MDISK 29A 3350 124 008 MAINT WR RMAINT WMAINT MMAINT
  MDISK 29B 3350 132 020 MAINT WR RMAINT WMAINT MMAINT
  MDISK 29C 3350 152 003 MAINT WR RMAINT WMAINT MMAINT
  MDISK 29D 3350 155 008 MAINT WR RMAINT WMAINT MMAINT
```

Notes:

1. *Your directory will contain additional directory statements where the ellipsis (:) appears.*
2. *If you use the High Performance Option (HPO), you may want to omit DIAG98 on the OPTION statement. See "Real I/O in HPO Systems" on page 38 for an explanation.*

Figure 11. Sample Directory Entries for the MAINT Userid

```
USER VTAM VTAM 10M 16M ABG
  OPTION ECMODE DIAG98 MAXCONN 400
  IUCV *CCS P M 10
  IUCV ANY P M 0
  ACCOUNT VTAM GCS
  IPL GCS PARM AUTOLOG
  CONSOLE 01F 3215
  SPOOL 00C 2540 READER A
  SPOOL 00D 2540 PUNCH A
  SPOOL 00E 1403 A
  LINK MAINT 190 190 RR
  LINK MAINT 595 595 RR
  LINK MAINT 298 191 WR
  LINK MAINT 29A 29A RR
```

Figure 12. Sample Directory for the VTAM Userid

The Minidisks You Will Need

VTAM requires several minidisks for installation and service. Do not change the default addresses for these minidisks unless you are already using the same addresses for another purpose. If you make changes, you must update the 5664280 VMFPARM file in Step 2 on page 43. (You will find a description of this file in "The 5664280 VMFPARM File" on page 39.)

These addresses apply to the MAINT userid, and all these minidisks should be owned by MAINT.

Function of Disk	**Disk Address**	**Size Required (in 1K Blocks)**
CMS system disk[1]	190	not applicable[3]
MAINT's 191 disk[1]	191	not applicable[3]
IPCSE disk[1]	193	40[2]
TRAPRED disk[1]	193	20[2]
Work space for INSTFPP[1]	194	not applicable[3]
GCS system disk[1]	595	not applicable[3]
MACRO disk[1]	19E	1800[2]
VTM191 disk	298	1350
BASE disk	299	9000
RUN disk	29A	3600
MERGE disk	29B	9000
ZAP disk	29C	1350
DELTA disk	29D	3600
MNT319 disk[1]	319	not applicable[3]
VMFPARM disk[1]	348	15[2]

Notes:

1. *These disks must already be defined to VM before you install VTAM. VTAM should not share any disks except these.*

2. *VTAM shares these disks with other programs. The numbers represent the number of 1K blocks you should assign besides the space that is dedicated to other uses.*

3. *These are VM system disks; there are no size considerations for VTAM.*

The VMFPARM disk must be a separate disk. You cannot define it as a disk on which data will be written during the installation or service process.

Because all disks are owned by the maintenance userid, you do not need to update any EXECs or files when you change CP directory passwords.

Privilege Class

The VTAM userid only needs to be defined as a class G user. However, you may find classes A and B useful because they allow CP commands such as ATTACH, ENABLE, and SET FAVORED to be issued from the VTAM virtual machine.

Virtual Machine Size

The minimum recommended VTAM virtual machine size is 2M. The maximum size is 16M. The size should not be higher than the lowest address of the VTAM discontiguous shared segment (DCSS). The sample directory entry sets a default size of 10M to accommodate the GCS shared segment, which is defined as 2M in size starting at the 8M address. Since the VTAM DCSS starts at the 10M address, the VTAM virtual machine does not overlap the storage defined for the DCSS.

I/O Considerations

Real I/O

The DIAG98 option in Figure 12 on page 36 allows VTAM to use "real I/O" operations. This is strongly recommended if you are not using the VM/SP High Performance Option (HPO). "Real I/O" allows the VTAM virtual machine to fix (or "lock") pages in CP storage and execute channel programs that contain CP storage addresses. These addresses do not require translation, which improves VTAM's performance. This is because channel command words (CCWs) are not translated by CP, but are real addresses in the VTAM virtual machine.

CP restricts the use of real I/O page frames from a discontiguous shared segment. The page frames can only be locked if the shared segment is defined within the virtual machine's storage; that is, the virtual machine size must be large enough to overlap the shared segment. To use "real I/O," you should set the VTAM directory entry default size so that it overlaps the GCS saved system. The GCS recovery machine should also be large enough to overlap the GCS shared segment so that pages can be unlocked in case the VTAM machine is reset. Figure 10 on page 33 is an example of how you can do this.

Real I/O in HPO Systems

If you use HPO, although it supports processors with real storage greater than 16 megabytes, the use of DIAG98 requires "real I/O" to take place only in the first 16M of real storage. Using DIAG98 in these large processors requires that the pages containing VTAM buffers be fixed below the 16M real storage address; as a result it could cause contention for storage.

Specifically, you are trading the benefits of CCW translation against real storage contention and induced page stealing below the 16M address. The best approach is to run without DIAG98 and observe the demand for this storage by other system resources. If the demand is low, try running with DIAG98 and see whether the system's performance improves.

See *VTAM Customization* for more information on tuning VTAM in a VM environment.

Required Options

The IUCV *CCS option in Figure 12 on page 36 allows VSCS to communicate with CP. This option is required for VSCS operation.

The IUCV ANY option allows GCS to communicate with CP. This option is required for GCS operation.

Installing VTAM

This section tells you how to install VTAM on a VM/SP system.

Before installing VTAM, log on to MAINT and:

1. Define storage as 12M.
2. IPL CMSL.
3. Attach the product tape at address 181.
4. Be sure the VMFPARM disk (address 348) is accessed with a filemode between O and Z.

Do *not* access the RUN, ZAP, MERGE, DELTA or BASE disks. They will be accessed in the proper search order by the installation EXECs.

The EXEC you invoke to begin the installation procedure, INSTFPP, must be on MAINT's 191 disk. INSTFPP uses the disk at address 319 for product information and the disk at address 194 for work space.

Note: If you do the installation from a userid other than MAINT, you must establish write access to the minidisks before beginning the installation procedure.

The 5664280 VMFPARM File

The file 5664280 VMFPARM is used during the installation process. It contains information about minidisks that VTAM uses and about the VTAM discontiguous shared segment. Figure 13 on page 40 shows what the file looks like.

Each entry in the file, except the last one, applies to a separate minidisk and has two or more parts:

disk – name
Is the label used when the installation EXECs format a disk. Do *not* change this name in the 5664280 VMFPARM file.

primary – disk – address
Is the address used when the installation EXECs load the VTAM product files onto the disk. During the installation process, or any time after, you can use XEDIT to edit 5664280 VMFPARM file and change this address to PASS. If you do so, the disk is skipped during installation; no files are put onto it. You can code the primary disk

address as PASS and provide secondary addresses to avoid modifying a disk during an installation or service procedure.

secondary – disk – address(es)
Are optional parameters used to allow additional disk names for disk accesses. Use these addresses *only* for the BASE, ZAP, and MERGE disks. They allow you to verify service (such as PTFs) and back it out if necessary, before you actually merge it onto your system.

The last entry in the file, VTAMSEG, contains the name of the VTAM discontiguous shared segment. This name must match the name given to the VTAM discontiguous shared segment when it was defined in the CP system name table (see "Generating the Control Program (CP)" on page 34) and is the same name used when generating GCS. The default name is VTAM.

```
BASE    299
***     VTAM Base disk
***     VTAM files are installed onto this disk.

DELTA   29D
***     VTAM PTF staging disk
***     VTAM service is placed onto this disk.
***     The VMFMERGE function is used to move service from
***     this disk onto the MERGE disk.

IPCSE   193
***     IPCSE library disk
***     The VTAM dump formatting routines are put on this disk.
***     The VSCS dump formatting routines are put on this disk.

TRAPRED 193
***     TRAPRED library disk
***     The VTAM trace formatting routine is put on this disk.
***     The VSCS trace formatting routine is put on this disk.

MACRO   19E
***     VTAM macro library disk
***     The VTAM macro library is generated on this disk.
***     After the VTAM product has been installed, we recommend that
***     this disk be specified as PASS.

MERGE   29B
***     VTAM Merged text disk
***     VTAM service is merged onto this disk.
***     This disk is also used to load merged service.
***     This contains text files for modules that have changed
***     (due to service) since VTAM was installed.

RUN     29A
***     VTAM System library disk
***     The VTAM Load library is built on this disk.
***     The VSCS Load library is built on this disk.
***     This disk must be specified, and cannot be PASS.
```

Figure 13 (Part 1 of 2). The Default 5664280 VMFPARM File

```
VMFPARM  348
***    VMFPARM disk
***    This file (5664280 VMFPARM) is put onto this disk during
***    the installation procedure.
***    This disk address should be specified as a disk that is
***    currently accessed (during installation) and that will always
***    be accessed during installation or service processing.

VTM191   298
***    VTAM's 191 disk
***    Sample EXECs and source modules are put on this disk.
***    (But only during installation.  After installation, this disk
***    is not modified.)

ZAP      29C
***    VTAM ZAP target disk
***    Any local ZAPs are put on this disk.

VTAMSEG  VTAM
***    This is not a disk address, but is the name of the VTAM
***    Discontiguous Shared Segment.  If this parameter is modified
***    GCS must also be modified to load the specified segment.
```

Figure 13 (Part 2 of 2). The Default 5664280 VMFPARM File

Run-Time System

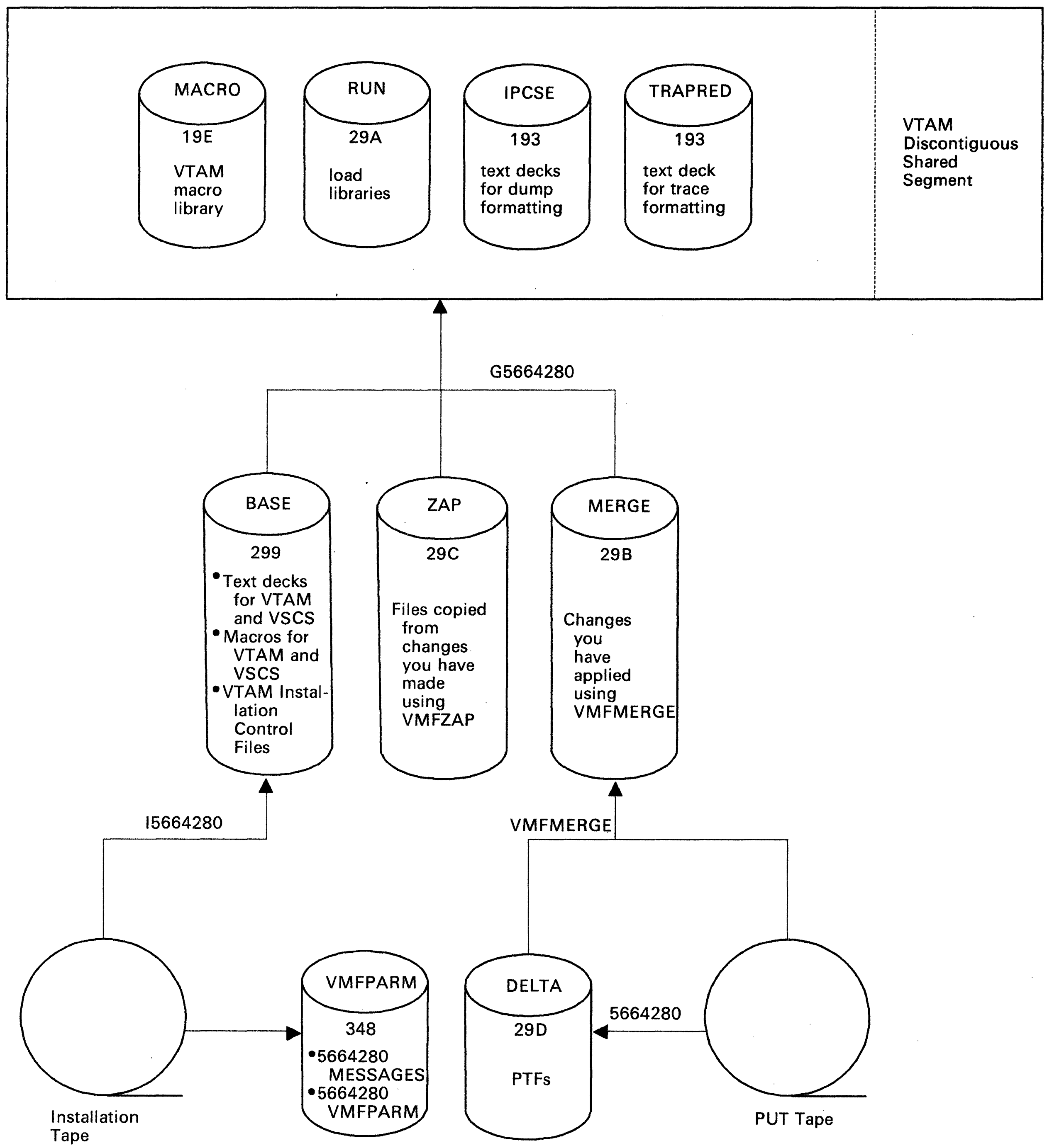


Figure 14. What Happens When You Install and Maintain VTAM on a VM System

The Installation Procedure

You install and maintain VTAM on a VM system by using a series of EXECs. Figure 14 on page 42 shows what happens.

To begin the installation, issue the command **INSTFPP 5664280**. This calls INSTFPP EXEC, which transfers the following items from tape to minidisk:

- I5664280 EXEC, the product installation EXEC
- Control files needed during the installation process
- Special documentation for the installer.

INSTFPP then invokes I5664280 EXEC, which does the following:

1. Asks whether you want to begin the installation process. Respond YES if you are ready to begin.

2. Invokes XEDIT so you can update the control file 5664280 VMFPARM. Do not change the default addresses for these minidisks unless you are already using the same addresses for another purpose. If you change the addresses in 5664280 VMFPARM, remember that you probably will need to update other files as well, such as the directory for MAINT.

3. Asks whether you want to continue the installation process. Respond YES if you are ready to complete the installation.

4. Loads the BASE disk, the MERGE disk, and the DELTA disk from the installation tape.

   I5664280 EXEC formats only those disks that are not already accessed by MAINT. The installation procedure does not erase anything on any disk, but it overlays files that have the same names as those created by the procedure.

5. Invokes G5664280 EXEC, the product generation EXEC. This EXEC copies files onto the appropriate minidisks, creates the VTAM user macro library, creates the VTAM load libraries and generates the discontiguous shared segment needed by VTAM.

Contents of VTAM Minidisks and Shared Segment after the Installation

When you finish installing VTAM, the minidisks have the following information on them:

The **VMFPARM (348)** disk contains:

- 5664280 MESSAGES, a sample VTAM messages file
- 5664280 VMFPARM, a sample VTAM parameter file.

These are control files required by installation and service EXECs. Note that VTAM shares this minidisk with other programs.

The **BASE (299)** disk contains:

- I5664280 MEMO, which summarizes the process of installing VTAM and VSCS
- All VTAM installation and service EXECs
- All VTAM text decks
- All VSCS text decks
- All VTAM interface macros
- All VSCS interface macros (DTIGEN)
- All VTAM sample materials
- All VTAM installation control files:
  - VTAM LKEDCTRL, a VMFLKED control file
  - VSCS LKEDCTRL, a VMFLKED control file
  - VTAM DCSSCTRL, a VTMDCSS control file
  - VTAMAC EXEC, a VMFMAC control file
  - VTAMGCS FILELIST, a list of all sample GCS EXECs for VTAM
  - VTAMLST FILELIST, a list of all sample VTAMLST files for VTAM
  - VTAMASM FILELIST, a list of all sample ASSEMBLE files for VTAM and VSCS
  - VTAMTXT FILELIST, a list of all TEXT files for VTAM and VSCS.

The **IPCSE (193)** disk contains VTAM text decks required for formatting GCS dumps with the Interactive Problem Control System (IPCS). Note that VTAM shares this minidisk with other programs. (For more information about IPCS, see the *VM/SP Interactive Problem Control System Guide.*)

The **MACRO (19E)** disk contains VTAMAC MACLIB. This macro library is used when assembling VTAM application programs under CMS. *This disk should be the standard system disk for general users.* Note that VTAM shares this minidisk with other programs.

The **RUN (29A)** disk contains:

- VTAM LOADLIB, the VTAM load library
- VSCS LOADLIB, the VSCS load library
- VTAM LKEDIT, the VTAM load library link-edit map
- VSCS LKEDIT, the VSCS load library link-edit map
- 5664280 VMFSVLOG, an empty initial service log file
- VTAM MAP, the VTAM discontiguous shared segment map.

The **TRAPRED (193)** disk contains a text deck needed to format VTAM trace records created by CPTRAP. For more about CPTRAP, refer to the *VTAM Diagnosis Guide.* Note that VTAM shares this minidisk with other programs.

The **VTM191 (298)** disk contains sample installation material for VTAM, including:

- ISTAPPLS VTAMLST, definitions of application programs
- ATCCON00 VTAMLST, definitions of system resources
- ATCSTR00 VTAMLST, start parameters for VTAM
- VMVTAM GCS, a sample EXEC for starting VTAM
- ISTINCDT ASSEMBLE, a default USS table for VTAM

- ISTINCLM ASSEMBLE, a default logon mode table for VTAM and NCCF
- ISTINCNO ASSEMBLE, a default USS table for VTAM operator commands and messages.

Note: After the installation, you should also place all NCP major node definitions on VTM191.

The **ZAP (29C)** disk is formatted, but it is empty. It is provided as a place for ZAP fixes to be placed.

The **MERGE (29B)** disk contains 5664280 VMFMGLOG, an empty initial service log file. This disk will eventually contain fixes for text files on the VTAM BASE disk.

The **DELTA (29D)** disk contains a file named DUMMY FILE. During service processing, VTAM fix packages (PTFs) are put on this disk. These fixes are installed on the MERGE disk using VMFMERGE.

The **MNT319 (319)** disk contains files used by INSTFPP during the installation process:

- 5664280 MEMO, which summarizes the process of installing service for VTAM and VSCS
- PROD LEVEL, which has information about the status of your system.

Note that VTAM shares this minidisk with other programs.

Verifying the Installation

To verify that the installation is complete, follow the following procedure.

Note: The VMVTAM EXEC, invoked in Step 2 on page 46, assumes that:

- *You have not changed the default address of the RUN disk (29A). (If you have changed it, you need to change the ACCESS command in VMVTAM EXEC.)*
- *You are using the VTAM default directory entry.*
- *The GCS recovery machine is already logged on.*
- *The VTAM machine is an authorized Group Control System (GCS) machine.*
- *You have issued the CP ENABLE SNA command.*

1. Log on to the VTAM userid. If you coded an IPL statement in the VTAM directory, an initial program load (IPL) is done automatically for GCS. If not, issue the IPL GCS command.

2. Enter VMVTAM to invoke the sample VTAM startup EXEC. You should receive these messages:

```
CSIACC423I F (29A) R/O
IST315I VTAM INTERNAL TRACE ACTIVE - MODE = INT, SIZE = 002,
     OPTIONS = API PIU MSG
IST093I APPLIST ACTIVE
IST020I VTAM INITIALIZATION COMPLETE
DTII10I DEFAULT INITIALIZATION PARAMETERS BEING USED
DTIC01I DTICINIT VSCS IUCV INITIALIZATION IS IN PROGRESS
DTIS74I TRACE TABLE STARTS AT address , ENDS AT address
DTIC02I DTICINIT VSCS IUCV INITIALIZATION COMPLETE
R;
DTIV04I DTIVINIT VSCS VTAM SERVICES INITIALIZATION COMPLETED
```

Message IST020I tells you that VTAM is installed and operational. The last message, DTIV04I, tells you that VSCS is installed and operational.

When you have finished this procedure, follow the instructions in "Coding the PROFILEs for the Virtual Machines." Then you can use the definition statements in Chapter 5, "Defining the Network to VTAM" on page 69 to define resources to VTAM and the information in *VTAM Customization* to customize VTAM.

To minimize the need to re-create problems that might be encountered during testing, it is recommended that you turn on the VTAM, VSCS, and GCS internal traces. However, in a production environment, these traces, if you leave them on, could affect VTAM's performance.

Coding the PROFILEs for the Virtual Machines

The examples under "Setting up Directory Entries for VTAM and MAINT" on page 35 give you information on defining the virtual machines in a GCS group to VM. The examples that follow show you how to code PROFILEs for the virtual machines so that VTAM and VSCS can be automatically started at system IPL time.

The following examples illustrate commands you could code in the PROFILEs for the AUTOLOG1, RECOVERY, and VTAM virtual machines.

The commands shown in the examples are those required to start VTAM. They should be in the PROFILEs along with any other commands you wish to have.

Note: For the RECOVERY and VTAM virtual machines, the filetype of the EXEC is GCS (not EXEC), because these virtual machines IPL GCS rather than CMS.

For details on the commands, refer to:

- *VTAM Operation* for the VTAM START and VSCS START commands
- *Group Control System Guide* for the GLOBAL and LOADCMD commands
- *VM/SP Operator's Guide* for the rest of the commands.

The PROFILE EXEC for the AUTOLOG1 Virtual Machine

Here is a sample PROFILE EXEC for the AUTOLOG1 virtual machine, which automatically starts the GCS recovery machine:

```
/*  */
CP AUTOLOG RECOVERY RECPASS
CP LOGOFF
```

AUTOLOG
Starts the RECOVERY virtual machine. **RECPASS** is the logon password for the RECOVERY virtual machine. You can use any password you want instead of **RECPASS**.

LOGOFF
Terminates the AUTOLOG1 virtual machine. The AUTOLOG1 virtual machine is not needed after the GCS recovery virtual machine is started.

The PROFILE GCS for the GCS Recovery Virtual Machine

The VTAM virtual machine is started by the GCS recovery virtual machine, rather than by the AUTOLOG1 virtual machine, because the recovery virtual machine must be the first machine started in a GCS group.

Here is a sample PROFILE GCS for the GCS recovery virtual machine, which automatically starts the VTAM virtual machine:

```
/*  */
CP AUTOLOG VTAM VTAM
CP SET FAVORED RECOVERY 100
CP SET FAVORED RECOVERY
CP SET PRIORITY RECOVERY 1
```

AUTOLOG
Starts the VTAM virtual machine. **VTAM** is the password you coded on the USER control statement in the VM directory for the VTAM virtual machine. You can use any password you want instead of **VTAM**, provided it matches the password in the directory.

SET FAVORED
Improves the performance of GCS during error recovery.

SET PRIORITY
Improves the performance of GCS during error recovery.

Note that the directory for the VTAM virtual machine automatically causes an IPL GCS command when VTAM is started. (The directory is shown in Figure 12 on page 36.)

For details on the syntax of these commands, see the *VM/SP Operator's Guide* or the *VM/SP HPO Operator's Guide*.

The PROFILE GCS for the VTAM and VSCS Virtual Machine

Figure 15 shows a sample PROFILE GCS for the virtual machine in which VTAM and VSCS run. At the end of the sample, you can add any commands you want.

```
CP SET FAVORED VTAM 80
CP SET FAVORED VTAM
CP SET PRIORITY VTAM 1
CP SET QDROP VTAM OFF
ACCESS 29A F/F
GLOBAL LOADLIB VTAM VSCS NCP SSPGCS USSTAB MODETAB
FILEDEF NCPLOAD DISK NCP LOADLIB A
LOADCMD VTAM ISTINV00
LOADCMD VSCS DTISLCMD
VTAM START LIST=01,...
VSCS START
CP ENABLE SNA
   :
   :
```

Note: If you use the High Performance Option, do not set QDROP OFF. Instead, use SET RESERVE and SET MINWS commands.

Figure 15. Sample PROFILE GCS for the VTAM Virtual Machine

SET FAVORED
Improves the performance of VTAM and the response time of SNA consoles running under VM.

SET PRIORITY
Improves the performance of VTAM and the response time of SNA consoles running under VM.

SET QDROP OFF
Prevents VM from dropping the VTAM virtual machine, when it is idle, from the active queue. Placing this command in the PROFILE GCS improves the performance of VTAM and the response time of SNA consoles running under VM.

If you use the VM/SP High Performance Option (HPO), do *not* set QDROP OFF. Instead, use these commands:

SET RESERVE VTAM nnnn
SET MINWS VTAM nnnn

The value you chose for *nnnn* should be the same for both commands.

GLOBAL LOADLIB
Identifies to the VTAM virtual machine the load module libraries required for VTAM and VSCS, for loading and dumping the NCP, for the USS table, and for the logon mode table.

FILEDEF
Defines the NCP load library called NCP LOADLIB A on the VTAM 191 disk. This example assumes that the DDNAME=NCPLOAD on the NCP's BUILD statement.

LOADCMD VTAM
Loads the VTAM initialization module and establishes "VTAM" as the prefix to be used for operator commands.

LOADCMD VSCS
Loads the VSCS initialization module and establishes "VSCS" as the prefix to be used for operator commands.

VTAM START
Starts VTAM in the virtual machine. *LIST=01* means the configuration list to be used. You may also include other options on the START command.

This command completes when VTAM initialization is complete and VTAM is ready to accept new commands.

VSCS START
Starts VSCS in the virtual machine.

ENABLE SNA
Enables Console Communication Services, so VSCS can use it to provide SNA console support.

ACCESS
Accesses the RUN (29A) minidisk. This disk contains load libraries for VTAM and VSCS.

Installing Service

Installing service—such as program temporary fixes (PTFs)—is similar to installing VTAM for the first time. A product update tape (PUT tape) contains a MERGE file (which contains text files with integrated service), and a DELTA file (which contains control files and text files). The VTAM service EXEC, 5664280, reads the files from the tape and calls an EXEC, G5664280, that generates the VTAM discontiguous shared segment and link-edits VTAM modules.

To install VTAM service on a VM system, log on to MAINT and follow this procedure:

1. Erase all files from your DELTA disk.

2. Make sure that the VTAM you are updating is not in use. The VTAM userid is *not* logged off automatically when you apply service.

3. Make sure that the VMFPARM (348) disk is accessed with a filemode between O and Z.

4. Make sure that the RUN, ZAP, MERGE, DELTA and BASE disks are *not* accessed. They will be accessed in the proper search order by the service EXECs.

5. Issue the command **VMSERV**. It asks you what kind of service you want to install. Depending on your answer, you will need to do one or more of these steps:

   - *To exclude PTFs*: using XEDIT, add to the file 5664280 EXCLIST the numbers of any PTFs you want to exclude. Add to the file 5664280 APPLIST the number of any excluded PTFs you want to apply.

   - *To move data onto the MERGE disk*: issue the command **VMFMERGE 5664280**. This updates your MERGE disk with the appropriate level of service.

   - *To refresh the contents of the ZAP disk*: issue the command **VMFZAP 5664280**. This reapplies ZAPs that you still need and deletes ZAPs you no longer need.

6. Issue the command **VMSERV BUILD 5664280**. This generates the updated VTAM system.

Reinstalling VTAM

The easiest way to reinstall VTAM is to repeat the process in "The Installation Procedure" on page 43. Before you reinstall VTAM:

- Access the VMFPARM disk with a filemode between O and Z.
- Be sure the VTAM you are replacing is not in use. The VTAM userid is *not* logged off automatically when you do the installation procedure.
- Erase all the files on the BASE and MERGE disks.

If you want to build the VTAM system without loading from tape, the 5664280 EXEC is on the VMFPARM disk. Access this disk as the C-disk, and issue 5664280 BUILD to rebuild the VTAM system. The 5664280 EXEC is initially installed on the BASE (299) disk.

During the installation process you can, if you wish, save copies of files that the process will modify. These include:

- 5664280 VMFPARM. When the installation process lets you use XEDIT to update this file (step 2 on page 43), you can copy it onto a minidisk that is not owned by VTAM.
- Sample installation materials. To save these, use PASS as the address of the VTM191 disk in the 5664280 VMFPARM file. Doing this prevents VTM191 from being updated by the installation process.

Chapter 4. VSE Installing VTAM in VSE

Overview of the Process

VTAM is an access method of the VSE system. Before VTAM can be used, it must be incorporated into the VSE system during system generation. Once VTAM has been identified to the system, it will not be necessary to again perform a system generation for this system when upgrading to a new release of VTAM. You will find information on planning and generating VTAM support in a VSE system in *VSE/Advanced Functions System Generation* and *VSE/Advanced Functions System Management Guide.*

The process of installing VTAM from the tape, including suggested coding statements, is described in the Memo to Users and Program Directories that accompany the VTAM product.

The Program Directory also describes the contents of the tape and space allocations required on system files. The VSE Maintain System History Program (MSHP) is used to install the VTAM modules from the product tapes and link-edit them. MSHP is also used to install the VTAM macros in the source statement library.

VTAM becomes a part of the operating system as a result of installing the contents of the VTAM tape. These are the VTAM modules and macros.

VTAM Version 3 requires VSE Version 2 Release 1 with modification level 3.

Figure 16 on page 54 illustrates the process of installing VTAM in VSE.

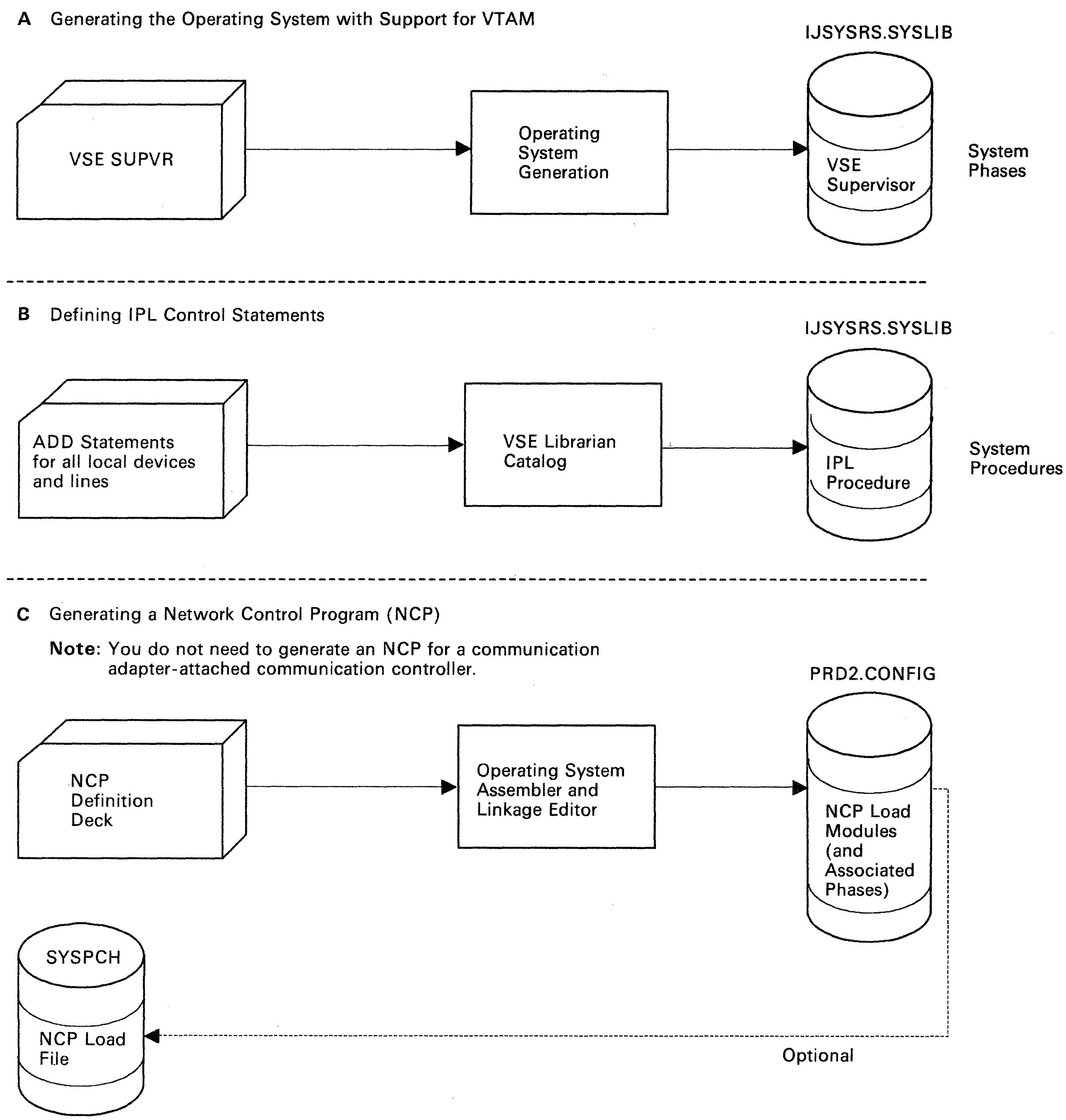


Figure 16 (Part 1 of 4). Creating a VTAM Domain in VSE

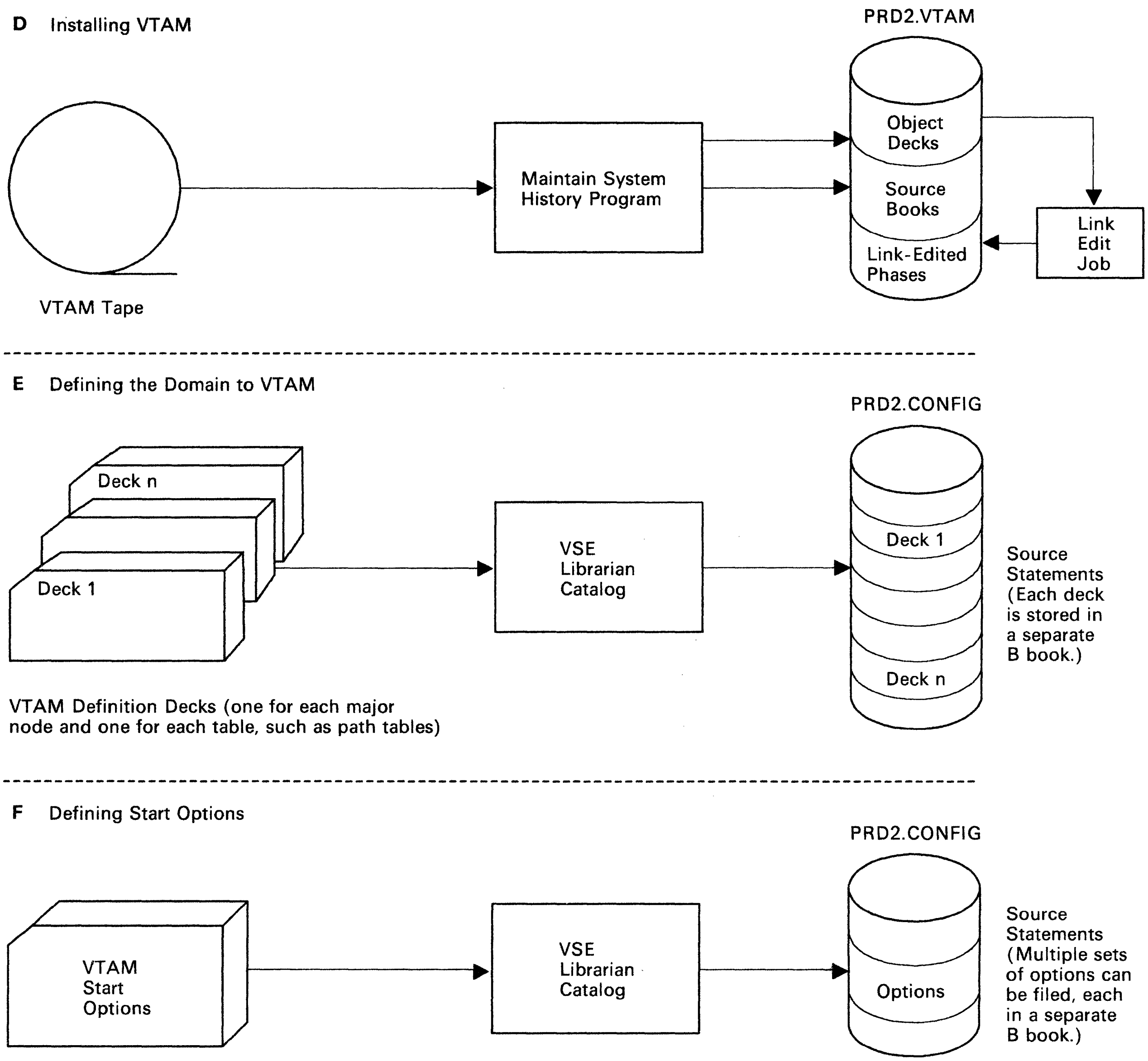
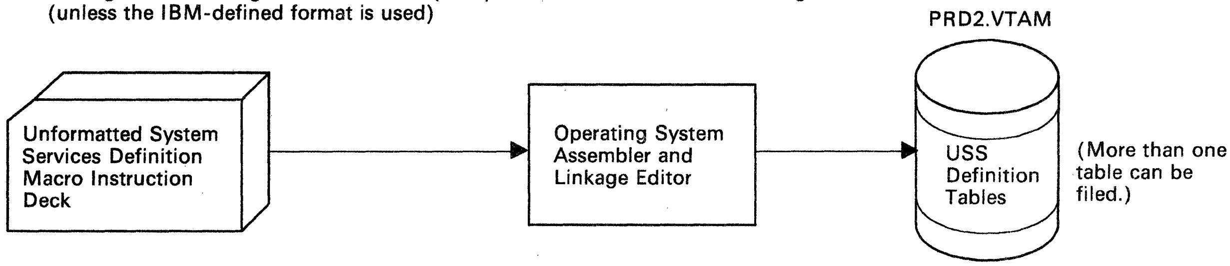


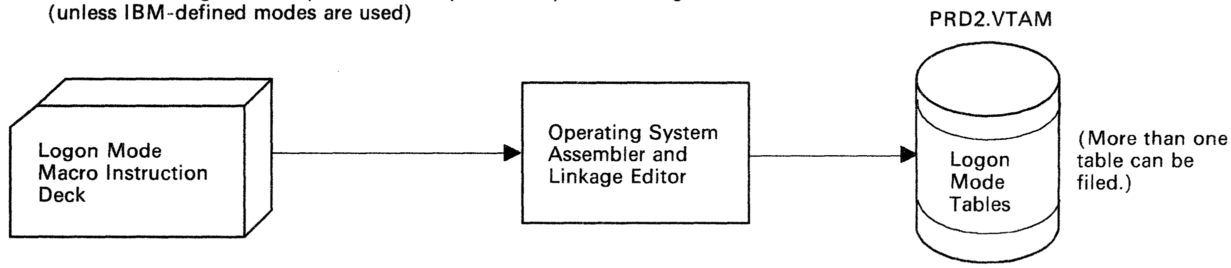
Figure 16 (Part 2 of 4). Creating a VTAM Domain in VSE

G Defining Session-Establishment and Termination Tables

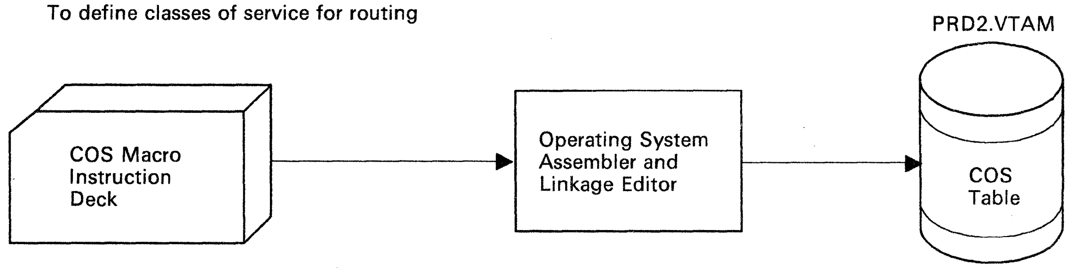
For logical units using character-coded (USS) commands and terminal messages (unless the IBM-defined format is used)



To define the logon mode (set of session parameters) for each logical lunit (unless IBM-defined modes are used)



To define classes of service for routing



For logical units using the interpret function

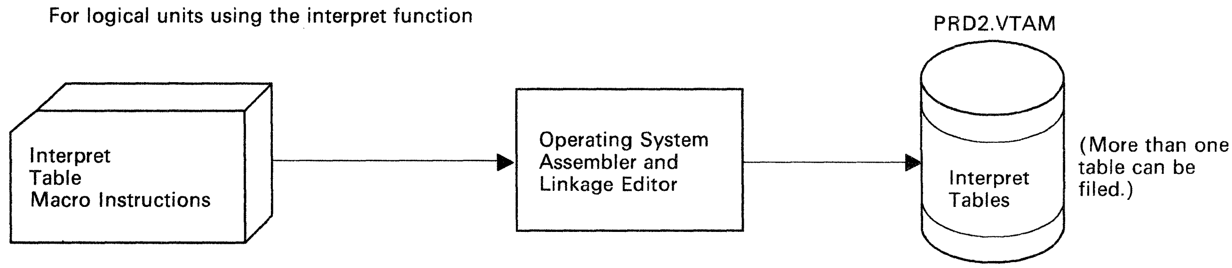


Figure 16 (Part 3 of 4). Creating a VTAM Domain in VSE

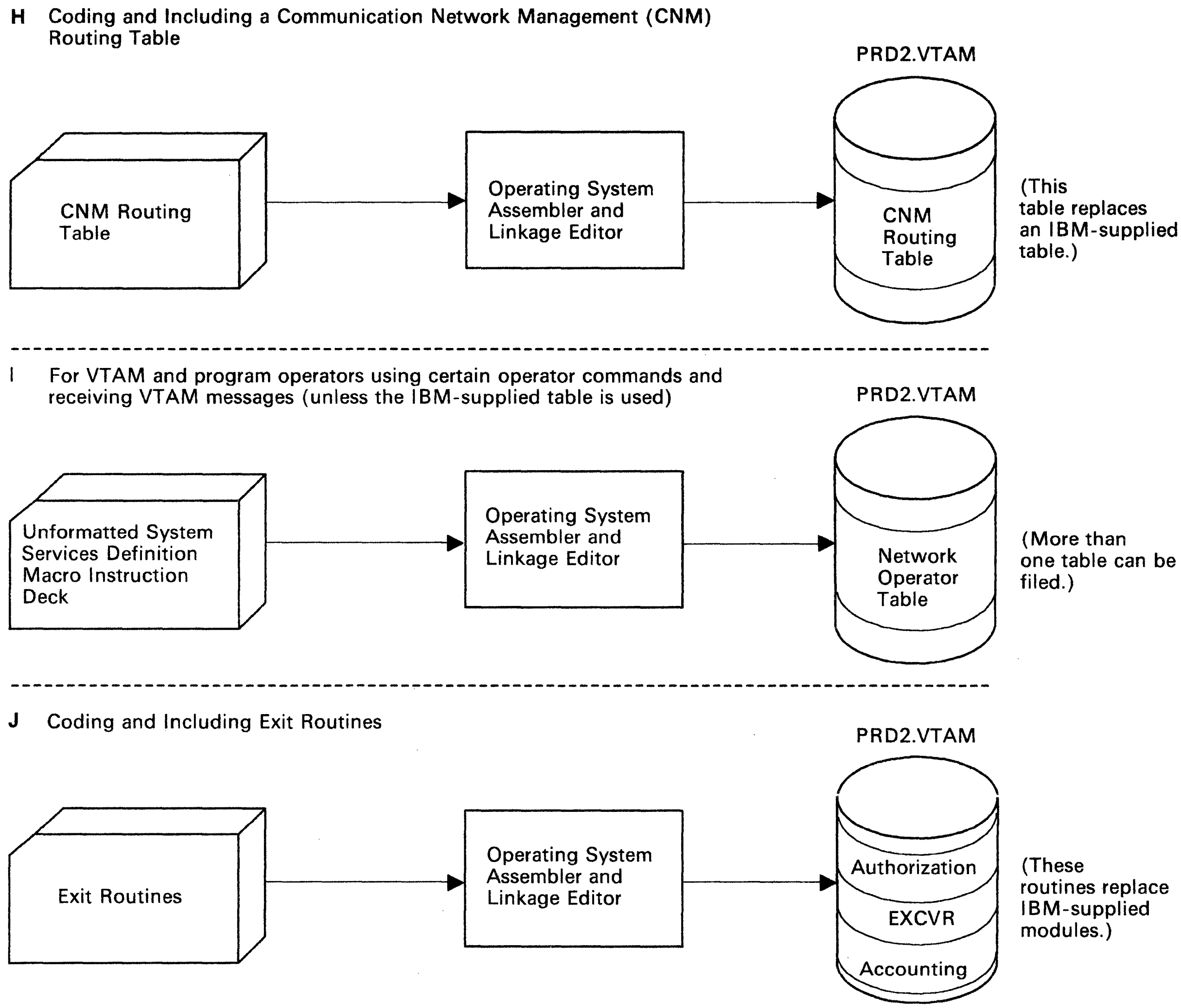


Figure 16 (Part 4 of 4). Creating a VTAM Domain in VSE

Notes:

1. *Although VTAM is shown as being installed on private libraries, it can be installed on either system or private libraries.*

2. *You do not need to generate an NCP for a 3705 or 3725 attached by a communication adapter.*

Identifying VTAM to the Operating System

You no longer need to identify VTAM to the operating system. The VSE system generation statements already include VTAM support. However, you can change values coded on some system generation statements, such as the SUPVR and IOTAB macro instructions.

The SUPVR Macro Instruction

A VSE system must have at least two partitions: one for VTAM and one for an application program. If additional partitions are needed (for other application programs or for programs unrelated to VTAM), change the NPARTS operand to include them.

The partition containing VTAM should have higher priority than partitions containing VTAM application programs; otherwise, the application programs might be not be able to communicate with VTAM. For example, a request to open an access method control block could fail. The relative priorities of the application programs are not significant to VTAM. Note that an application program can be a subtask of another application program running in the same partition.

The IOTAB Macro Instruction

The IOTAB (I/O Table) macro instruction defines the storage to be set aside for various system device tables. The IODEV operand determines the maximum number of devices that might be attached to the system. For VTAM, you must include all channel-attached terminals and all communication controllers attached directly to the host processor.

Coding an IPL Procedure with Support for VTAM (Optional)

An initial program load (IPL) procedure loads VSE and defines to VSE the devices that are to be attached to it and the sizes of the partitions it is to create. To include support for VTAM in an IPL procedure, you must define VTAM devices and define the sizes of VTAM-related partitions.

For an example of IPL statements, see Figure 17 on page 59.

```
ADD 01F,1050A
ADD 024,3705,01
ADD 025,3705,01
ADD 026,3705,01
ADD 027,3705,02
ADD 028,3705,02
ADD 030,3705,10
ADD 031,3705,10
ADD 032,3705,10
ADD 033,2703
ADD 034,2703
ADD 03B,2703
ADD 00A,3277
ADD 00B,3277
ADD 00C,3505
ADD 00D,3525P
ADD 00E,1403
ADD 080,3277
ADD 081,3277
ADD 082,3777
ADD 083,3277
ADD 090,3277
ADD 0B8,3791L
ADD 181,3420T9
ADD 182,3420T9
ADD 183,3420T9
ADD 190,2314
ADD 191,2314
ADD 192,2314
ADD 193,2314
ADD 194,2314
ADD 195,2314
ADD 196,2314
ADD 250,3330
ADD 251,3330
ADD 252,3330
ADD 253,3330
ADD 254,3330
DEF SYSREC=200,SYSCAT=200
DPD TYPE=N,UNIT=X'193',CYL=1,VOLID=111111
SVA
ALLOC R F1=200K,F2=118K,F3=0K,F4=0K,F5=0K,F6=0K
ALLOC F1=1750K,F2=250K,F3=0K,F4=0K,F5=0K,F6=0K
ASSGN SYSLST,02E
ASSGN SYS001,193
ASSGN SYS002,193
ASSGN SYS003,193
ASSGN SYSLNK,193
```

Figure 17. An Example of IPL Statements in VSE

Defining VTAM Devices

Insert an ADD command in the IPL procedure for each channel–attached device and each communication–adapter attached line. ADD commands relate device types to specific channel device names. For more information about the device type codes for specific devices, see the description of the ADD command in *VSE/Advanced Functions Operating Procedures.*

Defining Sizes of VTAM and Application Program Partitions

The sizes of VSE system partitions are defined during system generation by the ALLOC and ALLOC R operator commands. Both real and virtual storage must be allocated for the partition in which VTAM will run. You must allocate additional storage if a module is to be attached as a VTAM subtask. For information on calculating storage requirements, refer to *Network Program Products Planning.* VTAM does not need space in the application program partition.

Defining VSAM

You can use DEF SYSCAT=cuu to define the address of the VSAM master catalog volume. Note that ASSGN statements are not valid for SYSCAT as operator commands or job control.

For more information, refer to:

VSE/Advanced Functions Operating Procedures
VSE/Advanced Functions System Generation.

Files Used by VTAM

This section describes:

- VTAM–related files
- NCP–related files
- Trace files.

VTAM–Related Files

VTAM can use any of the following member types in its files:

- **VTAM definition library members**

  These members contain VTAM definition statements, start option lists, and configuration lists. They have the suffix **B** in their names. LCL3270.B is an example.

- **VTAM macro library members**

  These members contain VTAM macros. They have the suffix **E** in their names. OPNDST.E is an example.

- **VTAM phase library members**

  These members contain VTAM phases, tables, and exit routines. They have the suffix **PHASE** in their names. ISTCPC00.PHASE is an example.

- **VTAM object library members**

  These members contain VTAM object-code modules that have not been link-edited. They have the suffix **OBJ** in their names. ISTINCAL.OBJ is an example.

For more information on the library structure in VSE, see the *VSE/Advanced Functions System Management Guide.*

Configuration Restart VSAM Files

If the user chooses to use the configuration restart facility of VTAM, configuration restart VSAM files must be defined. See *Network Program Products Planning* for a discussion of configuration restart.

You can define configuration restart files for any of the following major nodes:

NCP major node
Channel-attachment major node
Local non-SNA major node
Local SNA major node
Switched major node
Channel-to-channel adapter major node.

For networks with more than one domain, you can define configuration restart files for these additional major nodes:

Cross-domain resource major node
Cross-domain resource manager major node.

Note: A different file is associated with each major node. When the definition of a major node is changed in the source statement library, you must delete it and redefine the associated configuration restart file.

A configuration restart file must have the following characteristics:

- The file must be indexed. Code the INDEX operand on the DEFINE command or take the default.

- A key length of 4 bytes and an offset of 0 bytes are required. Code KEYS (4 0).

- The average record size must be 24 bytes, and the maximum record must be 136 bytes. For example, for an NCP major node, code RECORDSIZE (24 136).

An example of access method service (AMS) statements that define a configuration restart file is shown below:

```
DEFINE-
  CLUSTER(NAME(ARNCKPT)-
          VOLUMES(111111)-
          KEYS(4 0)-
          RECORDSIZE(24 136))-
  DATA(NAME(ARNCKPT.DATA)-
          TRACKS(1))-
  INDEX(NAME(ARNCKPT.INDEX)-
          TRACKS(1))
```

The name of the configuration restart file in the DEFINE statement must match the name in the CONFGDS operand of the NCP PCCU statement or VBUILD or LBUILD statements.

The number of records in the file should equal the number of minor nodes defined in the major node. (For a switched major node, include each PATH statement when choosing the number of records.) Therefore, the primary allocation should be the number of minor nodes in the major node, and the secondary allocation should be about one tenth as big.

NODELST Files

You can define a NODELST file to maintain a list of major nodes and dynamic reconfiguration data set (DRDS) files that are active at one time. The system operator supplies the name of the file in the NODELST start option. If you choose to use the NODELST facility, one or more VSAM files must be defined. See *Network Program Products Planning* for a discussion of NODELST.

A NODELST file must have the following characteristics:

- It must be indexed. Code the INDEX operand on the DEFINE command or take the default.
- A key length of 2 bytes and an offset of 0 bytes are required. Code KEYS (2 0).
- The average record and the maximum record each must have a length of 10 bytes. Code RECORDSIZE (10 10).

The number of records in the file should equal the number of major node and DRDS file activations that occur from the time VTAM is started until it is halted. This includes major nodes that are reactivated after being deactivated. The primary allocation should be about 1.2 times the total number of major nodes and DRDS files in the network, and the secondary allocation should be about 0.2 times the total number.

Here is an example of statements that define a NODELST file:

```
DEFINE-
  CLUSTER(NAME(NODEDS1)-
          VOLUMES(111111)-
          KEYS(2 0)-
          RECORDSIZE(10 10))-
          SHAREOPTIONS(4))-
  DATA(NAME(NODEDS1.DATA)-
          TRACKS(1))-
  INDEX(NAME(NODEDS1.INDEX)-
          TRACKS(1))
```

The name of the node list file in the DEFINE statement must match the name in the DLBL statement in the VTAM start procedure. This name can be placed in a start option list in either the NODELST or CONFIG options.

NCP – Related Files

The files described below may be needed for a communication controller:

A load file that contains the NCP in card – image format or the PHASE file containing the NCP load module

A test routine file that contains the initial test routine of the NCP loader (one file for each host VTAM)

A dump file to receive the NCP dump output.

NCP Load File

VTAM can use either a phase or sequential data files as input to the NCP loader. The NCP stand – alone loader utility can use only sequential data files.

An NCP phase is loaded as one module and thus can be loaded faster than a sequential data file.

During Stage 3 of the generation of an NCP, the NCP load modules are automatically link – edited into a selected sublibrary. Unless you supply the name of a different sublibrary, PRD2.CONFIG is used. The phase name is the same as the name in the NEWNAME operand of the BUILD definition statement.

When the VTAM utility that loads NCPs is invoked to load this NCP, it searches for a copy first in the sublibrary you have referred to in the VTAM procedure. If the phase is found in PRD2.CONFIG (or in a different sublibrary, if you so specified) and enough storage has been allocated to the VTAM partition through the ALLOC statement to hold the phase, the phase is used. If there is not enough storage, VTAM looks for a copy of the NCP on a sequential data file.

If you have storage considerations or want to use the stand – alone utility, you should create sequential data files of your NCPs. To do this, punch the NCP phase to the sequential data file whose file name matches the name in the NEWNAME operand.

To use the VTAM utility for loading an NCP, you must add to your VTAM partition storage allocation an amount equal to the size of the largest NCP load module you expect to process. If you do not know its size, use list directory output to determine the correct value. It is recommended that you define the NCP load file as described below as a backup in case NCP generation changes or reallocation of operating system storage prevents use of this loading method.

VTAM uses some of the NCP utilities to load an NCP. The LIBDEF statement in the VTAM start procedure should have the file name of the sublibrary that contains the phases for the utilities.

The file name in the DLBL statement must be the same as the name in the NEWNAME operand of the BUILD definition statement. The symbolic device name in the EXTENT and ASSGN statements must be the same as the name in the NCPLUB operand of the PCCU statement. The BUILD and PCCU statements are discussed in Chapter 5, "Defining the Network to VTAM" on page 69.

You must provide JCL to punch the relocated NCP load image to a sequential file required by the communication controller loader utility and VTAM. SYSPCH can be assigned to a disk extent.

Refer to *NCP–SSP Installation* for more details.

Initial Test Routine File

For a channel–attached NCP, the VTAM user can load a diagnostic routine, called the initial test routine, into a communication controller and have it execute before the NCP is loaded. The initial test routine is distributed with the NCP generation tape. DLBL and EXTENT statements must be provided for this file if the routine is to be used. The file name in the DLBL statement must be DIAGFLE. The LUB must be SYS008.

NCP Dump File

The NCP dump utility writes the contents of the communication controller storage into a direct access device in 512–byte records. The file name in the DLBL statement must be NCPDUMP. The symbolic device name in the ASSGN statement must be the name in the DUMPDS operand of the PCCU definition statement. This symbolic device name must be omitted if more than one NCP is to be dumped. (The PCCU statement is discussed in "The PCCU Definition Statement" on page 171.)

The dump file must accommodate a dump of the entire communication controller storage. The size of communication controller storage depends on the model number.

The amount of direct–access storage required for the dump file can be determined from the storage size of the communication controller and from calculating the number of tracks needed to hold the required number of 512–byte records of communication controller storage.

VTAM dump processing fails if the NCP modules that are loaded to process the dump are not in the proper sublibrary or are in a different sublibrary during dump processing.

VTAM Trace Files

The trace information is written sequentially onto a disk or tape file. If the file is on a disk, it must be defined in DLBL and EXTENT statements.

For a disk trace file, the names TRFILE and SYS001 are mandatory. TRFILE must be the first operand of the DLBL statement.

For a tape trace file, assign SYS001 to an unlabeled tape.

To use TPRINT to print a trace, include SYS004.

When the disk 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 trace information onto SYSLST.

The trace records are written to TRFILE in 2048-byte blocks. The number of tracks of direct-access storage to be allocated depends on the number of 2048-byte records that the user wishes to preserve. The *VTAM Diagnosis Guide* contains more information about the traces available in VTAM.

Coding a VTAM Start Procedure

An EXEC statement begins the execution of VTAM. The user can create a procedure to start VTAM (called a start procedure) for EXEC, or the VTAM operator can enter the JCL from the console or a card reader. The VTAM operator can either enter the JCL in its entirety or modify a start procedure. The manner in which the operator issues EXEC, enters JCL, and invokes or modifies a start procedure is not unique for VTAM; these actions are done as described in *VSE/Advanced Functions Operation*. An example of the job control statements in a typical start procedure is shown in Figure 18 on page 66.

```
// JOB ACFVTAM
// ASSGN SYS000,UA                                              1
// LIBDEF PHASE,SEARCH=(PRD2.VTAM,PRD2.COMM,PRD2.COMM2)         2
*
// LIBDEF SOURCE,SEARCH=(PRD2.CONFIG)                           3
*
// ASSGN SYSLST,cua                                             4
*
// DLBL DIAGFLE,'IFUNCPD,IFUNCPE'                               5
// EXTENT SYS008
// ASSGN SYS008,cua
*
// DLBL NCP001,'NCP001.LOAD.FILE'
// EXTENT SYS007
// ASSGN SYS007,cua
*
// DLBL NCPDUMP,'NCPDUMP.FILE',,DA
// EXTENT ,,,,190,95
// ASSGN SYS005,DISK,VOL=VOLIDS,SHR
*
// DLBL NCPDUMP,'NCP.DUMPS'
// EXTENT ,VOLIDS
// ASSGN SYS005,DISK,VOL=VOLIDS,SHR
* OPTIONAL TRACE FILE
// DLBL TRFILE,'VTAM.TRACE.FILE'
// EXTENT SYS001
// ASSGN SYS001,cua
* OPTIONAL TPRINT FILE
// DLBL TRFILE,'VTAM.TRACE.FILE'
// EXTENT SYS004
// ASSGN SYS004,cua
* OPTIONAL CONFIGURATION RESTART FILE
// DLBL ANYNAME,,,VSAM
// EXTENT SYSCAT,VSAM01
* OPTIONAL CONFIGURATION RESTART FILE
// DLBL NCP001C,,,VSAM
// EXTENT SYSCAT,VSAM01
* OPTIONAL NODELST FILE
// DLBL NODELST,,,VSAM
// EXTENT SYSCAT,VSAM01
// EXEC ISTINCVT,SIZE=AUTO
*
```

Figure 18. Example of a Procedure to Start VTAM in VSE

The following notes are keyed to numbers in Figure 18.

Notes:

1. *Temporarily unassigned so VTAM can use the symbolic name SYS000 for the communication controller when loading an NCP. VTAM dynamically assigns SYS000 to the communication controller channel device name during LOAD and DUMP. SYS000 is also used when connecting or disconnecting channel–attached SNA cluster controllers.*

2. *VTAM phases can reside in a system library or a private library. You can provide job control language to name the private library to be searched before the system library. The file name of a private library must match the file name on the DLBL statement that identifies the private library. In this example, VTAM and NCP are in private libraries. The associated DLBL and EXTENT statements are in the standard labels for these libraries.*

3. *If the VTAM definition library is to reside in a private library, name the private library to be searched before the system library.*

4. *For the trace printout, SYSLST must be assigned. When the trace file is full, the trace print subtask of VTAM can be used to write the trace information onto SYSLST.*

5. *The names IFUNCPD and IFUNCPE are just arbitrary examples.*

Verifying the Installation

When VTAM has been installed, you can begin verifying it. See Chapter 8, "Verifying the Installation" on page 299 for suggested steps.

Chapter 5. Defining the Network to VTAM

If you are already familiar with the VTAM definition statements and operands and need only a quick reference, refer to Appendix A, "Quick Reference for VTAM Definition Statements" on page 313, an alphabetic reference that lists each VTAM definition statement and its operands.

If you are unfamiliar with the VTAM definition statements or need an explanation of an operand, refer to this chapter.

For VTAM to use a data communication network, the application programs and the physical configuration of the domain must be defined to it. These definitions represent points or nodes in the network that can be addressed and used by application programs (using VTAM macro instructions) and by the VTAM operator (using VTAM operator commands).

Defining the domain to VTAM involves defining:

- Application program major nodes
- Channel-attachment major nodes
- Local non-SNA major nodes
- Local SNA major nodes
- Switched major nodes
- NCP major nodes
- Sets of path statements.

Defining VTAM in a multiple-domain network also involves defining:

- Cross-domain resource manager major nodes
- Cross-domain resource major nodes.
- A default SSCP list within an adjacent SSCP table.

MVS VM Defining VTAM when you are using SNA network interconnection may also involve changing existing major node definitions.

After a network has been defined, peripheral SNA physical units (PUs) attached to an NCP and their logical units (LUs) in the NCP configuration can be added or relocated by defining dynamic reconfiguration decks.

Code definitions of major and minor nodes, paths, and adjacent SSCP tables using VTAM definition statements or NCP generation definition statements, and put these definitions into the VTAM definition library.

You must use network-unique names in all VTAM-related definitions that will be active at the same time. For example, the name on a statement defining a minor node must not be the major node name. A definition is termed "active" when VTAM reads the major node in which it is contained (see *Network Program Products Planning* for NCP backup exceptions). If major node or path definition set names appear in a configuration list, or if they are activated due to the VARY ACT operator command, VTAM will read them.

In addition, you cannot use any name beginning with "IST"—or any of the following names—for a major or minor node name:

VTAM

VTAMSEG

MVS VM VTAMTERM

VSE TRACE.

MVS You can code additional operands on some VTAM definition statements to use with the data encryption facility. These operands are described in Appendix D, "MVS Filing Cryptographic Keys" on page 339. If you plan to use the VTAM data encryption facility, you should first read *IBM Cryptographic Subsystem Concepts and Facilities.*

Introduction to Defining the Network

The following sections summarize:

- The procedure you must follow, and things you must consider, when defining the network to VTAM
- The format used in this chapter to describe the syntax of network definition statements.

Detailed descriptions of every definition statement begin at "Defining Session Pacing Values" on page 76 and continue for the rest of the chapter.

Filing Network Definition Statements

To define major nodes and file the definitions in the VTAM definition library, follow these steps. Follow them also when making changes or additions to the definitions after you have put them into the VTAM definition library.

1. Code the appropriate VTAM or NCP definition statements to define the major node. If defining an NCP:

   Follow the coding requirements for the NCP-only definition statements and operands and use the NCP generation instructions described in *NCP-SSP Resource Definition Reference* and *NCP-SSP Generation and Loading Guide.* If the NCP/EP Definition Facility is installed on your system, you may attach 3-letter prefixes to the NCP definition statements and operands described in this manual. However, attach prefixes only to statements included in an NCP major node definition. The results of doing otherwise are unpredictable.

If changing or adding VTAM – only definition statements or operands, change the NCP source deck; no NCP generation is required.

If changing or adding NCP definition statements or operands, change the NCP source deck and do a partial or complete NCP generation (as described in *NCP – SSP Installation and Resource Definition Guide).*

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

3. VM The network definition decks are stored as sequential CMS files with a filetype of VTAMLST. The filenames will be the names of the major nodes. The files are stored on a minidisk that is linked to the VTAM and VSCS virtual machine.

   Note: Whenever you change VTAM definition statements or start option lists, you must use the GCS ACCESS command to re – access the disk that contains the corresponding VTAMLST files before you activate the resources or restart VTAM.

4. Provide the VTAM operator with the necessary information to enter a VARY ACT command and respond with the proper information, if prompted by VTAM.

Sift – Down Effect in VTAM Definition Statements

The "sift – down" effect allows you to code an operand on a higher – level node so that you do not need to re – code it on each lower – level node for which the same value is desired. Thus, the sift – down effect can greatly simplify the coding process.

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 also override the "sifted" values for specific lower – level nodes by:

1. Coding the same operand on the lower – level node. If the operand is re – coded, the coded value applies until the next occurrence of the definition statement on which it is coded.

2. Coding the same operand with no value, followed by a comma (for example, SSCPFM = ,) will cause the default value to replace the sifted value for this and all lower – level nodes. However, if the next occurrence of a statement at the same level does not contain the operand, the original sifted value of the operand will be re – established.

The sifted values may be overridden at any point in the hierarchy of definition statements. Figure 19 on page 72 illustrates these sifting rules.

(The definition statements in the figure are indented to show their hierarchy.)

The specific keyword operands to which sifting applies are identified in the major node and dynamic reconfiguration definition sections in this chapter. For information about definition statement sequencing and the sift-down level for each NCP operand, refer to *NCP-SSP Installation and Resource Definition Guide.*

```
GROUP  SSCPFM=USSSCS
  LINE
    PU
      LU    (uses USSSCS)
  LINE   SSCPFM=USS3270
    PU
      LU    (uses USS3270)
  LINE   SSCPFM=USS3270
    PU    SSCPFM=,
      LU    (uses default value)
    PU
      LU    (uses USS3270)
  LINE
    PU
      LU    (uses USSSCS)
```

Figure 19. Example of Sifting Values

VM VM Considerations

All local device addresses that you code in definition statements for VTAM under VM must be *virtual* device addresses, rather than real device addresses. You must make sure that the virtual addresses that you code in the definition statements match:

- The virtual device addresses on any DEDICATE statements that you have coded in the VM directory
- The virtual device addresses on any CP ATTACH commands that you have coded for the PROFILE GCS EXEC for the VTAM and VSCS virtual machine
- The virtual device addresses on any CP ATTACH commands entered from the operator console
- The virtual device addresses on any CP DEFINE commands entered from the VTAM console or from a profile.

When you define a logical unit that will log on to VM, its name cannot be the same as any VM userids in the system.

Maximum Number of Minor Nodes in the Host Subarea

In application program major nodes, channel attachment major nodes, local SNA major nodes, and local non-SNA major nodes, some minor nodes may have to be counted as more than one node in calculating the maximum number of minor nodes in the host subarea. See *Network Program Products Planning* for more information.

Restrictions on Use of Assembler Features

The NCP and VTAM definition statements are coded in standard operating system macro instruction format, as described on page 74, with the following restrictions:

- Assembler program control instructions (such as ICTL, ISEQ) cannot be used in major node definition decks.

- Assembler listing control statements (such as PRINT, SPACE, EJECT) can be used in the NCP generation deck but must not be used in definition decks for the other types of major nodes.

Some assembler features must not be used in a major node definition deck:

- User assembler macro instructions that generate NCP definition statements are not permitted.

- Names generated by global variables (for example, &SYSNDX or &SYSECT) cannot be used.

- Variable substitution at assembly time is not permitted.

- References to assembler attributes (length, type, etc.) are not permitted.

- Use of literals is not permitted.

- Quoted strings cannot be used to make names out of keywords. For example, AUTH="PASS" is treated just like AUTH=PASS.

- Null keywords cannot be used. For example, ",," is treated as a null keyword and is invalid.

Comments, statements, or remarks can be used in decks for all the types of major nodes. However, macros or keywords, such as GENEND, etc., should not be used especially in a continued card of remarks.

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

Missing continuation characters can cause the NCP (during NCP generation) to assume values that are not physically correct (for example, half-duplex lines instead of full-duplex lines).

Format of Definition Statements in this Book

This section describes the conventions used in this book to explain the syntax of definition statements, and the rules used to code them.

The rules summarized here are assembler language rules.

NCP and VTAM definition statements have the following format:

Name	Definition Statement	Operands
Symbolic name	Definition Statement	Required and optional operands

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

1st character	uppercase alphabetic (A – Z) or the national characters @, #, or $
2nd to 8th	uppercase alphabetic (A – Z) or numeric (0 – 9) or the national characters @, #, or $

The name must begin in the first position of the definition statement and must be followed by one or more blanks. If the name is optional, you may still want to code it, since VTAM uses it in operator messages referring to the resource defined.

The **Definition Statement** field identifies the definition statement. It must be preceded and followed by one or more blanks.

The **Operands** field contains operands, which can be coded in any order and which must be separated by commas. The Operands field ends with one or more blanks placed after the last operand. In most definition statements, keyword operands are used in the Operands field. Keyword operands are followed by an equal sign (=) and the keyword value. The keyword 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.

You may code 3 – letter prefixes on NCP definition statements and operands if the NCP/EP Definition Facility is installed on your system. However, *do not* code prefixes on any other types of nodes. The results of doing so are unpredictable. For more information see *NCP – SSP Resource Definition Reference.*

If a default value for a keyword operand exists, you can cause VTAM to use it by doing one of the following:

- Omit the operand entirely. For example, code

    applname APPL

- Code a null value for the keyword. For example, code

  applname APPL MAXPVT=

  or, if other operands follow the defaulted operand, code

  applname APPL MAXPVT=,SONSCIP=YES

  All three examples have the same affect on MAXPVT as coding the operand with its default value, that is

  applname APPL MAXPVT=0

Comments can be written after the Operands field, but they must be separated from the last operand of the Operands field by one or more blanks. Comments can be continued to the next card, but they must start in column 17 or beyond. An entire card can be used for a comment by placing an asterisk in the first column of the card. If several entire cards are needed for comments, place an asterisk in the first column of each card and leave column 72 blank. A definition statement that has no operands cannot have comments on the same card as the operation code.

VTAM and NCP definition statements are coded in columns 1 through 71 of a card. You can continue a statement that exceeds 71 columns on one or more additional cards by placing a nonblank character in column 72. The operands can be interrupted either at column 71 or after any comma that separates operands. The continued portion must begin in column 16 of the following card. Comments can appear on every card of a continued statement. Columns 73 through 80 can be used to code identification characters, statement sequence characters, or both.

Symbols Used in this Book

This section lists the conventions used in this book to illustrate the format and coding of definition statements.

Capital Letters: Capital letters represent values that are coded directly, without change. Brackets [ ], "or" bar |, underlines, and subscripts are never coded.

Lowercase Letters: Lowercase letters represent operands for which a value or name must be supplied if the operand is coded.

Brackets [ ]: Brackets enclose operands or symbols that are either optional or conditional. Conversely, the lack of brackets means that you must code an item or group of items.

An optional operand is one that can be coded or omitted independently of other operands that are coded or omitted. Depending on the operand, omitting it might cause the corresponding feature or function to be omitted or included, or omitting it can cause a specific value (the default value) to be assigned. When the syntax of a definition statement is shown in this book, any operands that are required appear first, followed by the optional or conditional operands in alphabetical order.

Vertical "or" Bar (|): A vertical bar between operands means that you must choose one operand from among the values separated by the "or" bar.

Parentheses, Equal Signs, and Commas: Parentheses, equal signs, and commas are coded as shown.

Underlined Values: An underlined value represents the value that VTAM or the NCP uses if the operand is omitted (the default value).

Braces { }: Braces denote mutually exclusive operands.

Defining Session Pacing Values

There are two types of pacing: route pacing and session pacing. You can affect how VTAM performs route pacing through the PATH statement (discussed later in this chapter) or the VR pacing window size calculation module (see *VTAM Customization*).

You can specify session pacing in a logon mode table entry that is derived from the logon that requests the session (see *VTAM Customization*). You can also specify it on the statement that defines a logical unit to VTAM (for example, the PACING and VPACING operands on an LU statement).

An introductory description of pacing in *Network Program Products Planning* explains one- and two-stage pacing and how they relate to pacing counts. Figure 20 on page 77 summarizes how the PACING and VPACING operands relate to the various pacing stages for current VTAM and NCP operations. In this figure, PR and PS refer to primary receive and primary send counts; SR and SS refer to the secondary receive and secondary send counts.

Secondary-to-Primary Pacing

The primary receive count equals the VPACING value on the APPL definition statement, unless the secondary send count is zero (the secondary send count is found in the logon mode table entry associated with that session). If the secondary send count is set to zero, the primary receive count is also set to zero. This can be used to suppress secondary-to-primary pacing and may be required for devices that do not support secondary-to-primary pacing.

If the secondary LU (SLU) is in the same domain as the primary LU (PLU), the secondary send count equals the primary receive count, because only one-stage (same domain) secondary-to-primary pacing is supported.

If the SLU is a local SNA LU attached to a VTAM in another domain, two-stage pacing is supported. The primary receive count equals the VPACING value on the APPL definition statement. The secondary send count is obtained from the appropriate field in the CINIT RU.

Primary – to – Secondary Pacing

The secondary receive count is found in the logon mode table entry, unless that value is zero. If you code zero, the appropriate PACING or VPACING value of the secondary application program or LU is supplied. Figure 20 shows how the secondary receive count is determined for a specific secondary end.

The primary send count is found in the logon mode table entry, unless that value is zero. If you code zero, the VPACING value associated with the secondary application program or LU is supplied. Figure 20 shows how the primary send count is determined for a specific secondary end.

VTAM Definition Statement	Meaning of VPACING	Meaning of PACING	What the Operands Control
APPL	PR and SS	not allowed	One stage, secondary to primary, if the application program is primary.
APPL	PR	not allowed	Two stage, secondary to primary, if the application program is the primary and is in a cross domain session with a logical unit in a local SNA major node.[1]
APPL	PS and SR	not allowed	One stage, primary to secondary, if the application program is the secondary.
LU in local SNA major node	PS	SR[2]	One stage primary to secondary for same domain sessions. Two stage, primary to secondary for cross – domain sessions.
LU in switched major node	PS	SR	Two stage, primary to secondary.
LU in NCP major node	PS	SR	Two stage, primary to secondary.
LOCAL in local non – SNA major node	not allowed	not allowed	One stage, primary to secondary. (Can be specified by logon mode table.[3])
TERMINAL in NCP major node	not allowed	not allowed	One stage, primary to secondary. (Can be specified by logon mode table.[3])

Figure 20. Selecting Pacing Counts during Network Definition

Legend

PR primary receive count
PS primary send count
SR secondary receive count
SS secondary send count

Notes on Figure 20:

1. The SS value is defined by the SSNDPAC operand on the MODEENT macro instruction.

2. If both ends of the session are in the same domain, VTAM ignores the PACING operand. The PS and SR values are defined by the SRCVPAC operand on the MODEENT macro instruction used in the logon mode table to describe the session parameters. However, if that value is zero, the VPACING value is used for both the PS and SR values.

3. You can define the VPACING and PACING values in logon mode table entries. Three operands on the MODEENT macro instruction, which defines a logon mode table entry, can determine pacing counts: PSNDPAC is the PS count, SRCVPAC is the SR count, and SSNDPAC is the SS count.

   - Each resource can act as one end of a session with a VTAM application program.

   - You cannot select one-stage or two-stage pacing through network definition statements. VTAM makes this choice, based on the network configuration.

Non-SNA Pacing Counts

Cross-domain sessions with channel-attached non-SNA terminals can be paced. The pacing occurs between the primary LU in one host and the access method in the host that controls the channel-attached non-SNA terminals. Both primary-to-secondary and secondary-to-primary pacing are one-stage, and are always obtained from the logon mode table entry.

Overriding Defined Pacing Counts

The primary application program can override the primary receive value for a specific session by placing a nonzero value in the primary receive field of the session parameters it specifies at OPNDST. If the value is zero, the primary receive value present in the session parameters in the pending logon is used. The application program should specify only values less than or equal to the value in the pending logon.

The pacing counts in the definition statement for a logical unit apply to all sessions in which it participates. However, when you code AUTH=NVPACE in the definition statement for an application program in session with the logical unit, the counts defined by VPACING in the logical unit's definition statement are set to zero.

AUTH = VPACE is normally used to prevent overloading buffers in the communication controller with outbound messages from VTAM. However, you can use NVPACE in two situations:

1. When the application program sends only single-element request units (RUs) to any one LU and, after sending each RU, the program waits for a response before sending the next RU.

2. When the application program sends chains of RUs containing a limited number of elements and the program either (1) after sending one chain, waits for a response before it starts sending the next chain, or (2) sends the change-direction indicator as the last element of each chain. The number of elements in each chain must be no larger than the results of this formula:

   number of elements = $2n - 1$

   where n is the smallest value in the VPACING operands of the LU statements for the logical units.

If the application program is in session with logical units whose VPACING operands are different from each other, use the smallest VPACING values in the formula.

As an example of the calculation, if the LU statement contained VPACING = 3, the largest number of elements that should be sent by the application program in a chain would be:

$(2 * 3) - 1 = 5$

The pacing values in definition statements—PACING and VPACING—can be overridden for a specific session by the counts in the logon mode table entry used for that session (if the counts in the table are not set to zero). Finally, before actually establishing the session, an application program acting as a primary end can specify the primary receive count before it is sent in the BIND command.

Note: For information on how to code the PACING, VPACING, and AUTH = VPACE/NVPACE operands, refer to discussions of VTAM definition statements later in this chapter. For information on application programs specifying pacing information in session parameters, see *VTAM Programming*.

VSE VM Defining Loop-Adapter-Attached Devices

Loop-adapter-attached devices are defined in a local SNA major node and communicate with VTAM application programs as if they were channel-attached SNA physical units and logical units. See "Defining Local SNA Major Nodes" on page 143 for more information.

Defining Application Program Major Nodes

You define an application program major node by filing one VBUILD statement for the major node and an APPL definition statement for each application program in the major node.

See Figure 21 for an example of statements defining an application program major node.

```
APPLNODE VBUILD TYPE=APPL
APAYRO   APPL   ACBNAME=PAYROLL,AUTH=ACQ,PARSESS=YES,EAS=10,
                MODETAB=COMTAB
AACCTS   APPL   ACBNAME=ACCOUNTS,PARSESS=YES,EAS=20,
                MODETAB=COMTAB
AINVO    APPL   ACBNAME=INVENTRY,EAS=50
```

Figure 21. An Example of Statements Defining an Application Program Major Node

The VBUILD Statement

Name	Definition Statement	Operands
[name]	VBUILD	TYPE = APPL

name
is any symbol valid in the assembler language and is optional.

TYPE = APPL
means that this VBUILD statement defines an application program major node.

Application Program Major Node: APPL

The APPL Definition Statement

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

For information on coding the APPL definition statement for interconnected networks, see Chapter 7, "MVS VM SNA Network Interconnection Considerations" on page 247.

MVS If you are defining TSO/VTAM, refer to "Defining TCAS and Each TSO User" on page 326 for the TSO/VTAM requirements when coding the APPL definition statement.

Name	Definition Statement	Operands
name	APPL	[ACBNAME=acbname] [,AUTH=([ACQ|**NOACQ**] [,CNM|**NOCNM**] [,PASS|**NOPASS**] [,PPO|SPO|**NOPO**] [,TSO|**NOTSO**] MVS [,**VPACE**|NVPACE])] [,AUTHEXIT=(YES|**NO**)] VM [,DLOGMOD=default logmode entry name] [,EAS=n|**491**] [,ENCR=SEL|REQD|OPT|**NONE**] MVS [,HAVAIL=YES|**NO**] MVS/XA [,MAXPVT=**0**|n|nK|nM] MVS VM [,MODETAB=logon mode table name] [,PARSESS=YES|**NO**] [,PRTCT=password] [,SONSCIP=YES|**NO**] [,SPAN=(spanname)] [,SRBEXIT=YES|**NO**] MVS [,SSCPFM=USSNOP|**USSPOI**] [,USSTAB=name] [,VPACING=n|**0**] [,VTAMFRR=YES|**NO**] MVS

name

is a minor node name for this application program, as is the ACBNAME operand below. This name must be unique within the domain in which the application resides. Unless you use the Alias Name Translation function in NCCF and NetView, it must also be unique within the network if the application program is to engage in cross-domain sessions.

A logical unit using the application program's network name can log on to it from any domain in the network. A logical unit using the ACBNAME to log on to 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 macro instruction for this application program.

The ACB macro instruction is described in *VTAM Programming.*

ACBNAME = acbname

is the minor node name assigned to this application program. This name must be unique within the domain in which the application program resides. The name must also be the same as the name in the APPLID operand of the ACB macro instruction for this application program. If no application program name is coded in the APPLID operand of the ACB macro instruction, the name in the ACBNAME operand must be the job-step name. However, if the application is started under a job-step that invokes a procedure, the name in the ACBNAME operand must be the procedure-step name. This can be done for only one application program that is active within a job-step.

This operand is needed if (1) application programs with the same APPLID are activated in multiple hosts and the application programs wish to enter into a cross-domain session or (2) terminals in the network are to specify which application program in which host they wish to log on to.

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

MVS/XA For applications used in Extended Recovery Facility (XRF) sessions in MVS/XA, 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.

AUTH = ([operand][,operand]...)

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:

ACQ|<u>NOACQ</u>

determines whether this application program can use either the OPNDST macro instruction with the ACQUIRE option or the SIMLOGON macro instruction. (These macro instructions enable the application program to acquire a session with a particular logical unit.)

CNM|<u>NOCNM</u>

determines whether this application program can use the communication network management (CNM) interface. This interface allows an application program to send and receive formatted RUs from certain PUs in the application program's domain. For more information see *VTAM Programming.*

AUTH = CNM must be coded for all application programs for which entries exist in the IBM-supplied CNM routing table or for any user-written applications that use the CNM interface. For a complete list of IBM-supplied CNM applications, see *VTAM Customization.*

PASS|NOPASS

determines whether this application program can use the CLSDST macro instructions with the PASS option to pass session–establishment requests to other application programs.

For more information on the CLSDST macro instruction and simulating a logon (SIMLOGON), refer to *VTAM Programming.*

PPO|SPO|NOPO

is the status of the application program in regard to issuing VTAM operator commands and receiving responses and unsolicited messages. This facility is described in *VTAM Programming.*

If either PPO (primary program operator) or SPO (secondary program operator) is coded, the application program is authorized to issue SENDCMD and RCVCMD macro instructions. If this operand is omitted or if NOPO (no program operator) is coded, VTAM does not permit the application program to issue SENDCMD or RCVCMD macro instructions.

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

TSO|NOTSO MVS

tells VTAM 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.

VPACE|NVPACE

determines whether this application program is to be subject to the VPACING specifications of SLUs with which the program will be in session. Coding NVPACE is effectively the same as coding VPACING=0 in the LU statements for all of the SLUs with which the application program will be in session. NVPACE is ignored for same–domain local SNA LUs. See the pacing discussion earlier in this chapter and in *Network Program Products Planning.*

AUTHEXIT=YES|NO VM

tells VTAM whether the application program's exit routines are to run in supervisor state. If AUTHEXIT=YES, they will run in supervisor state even if the application is not authorized by GCS to do so. Before returning control to GCS, the application must clean up any recovery environment (ESTAE) it sets up.

The default is AUTHEXIT=NO.

DLOGMOD = default logon mode table entry name
is the name of the logon mode entry to be used by default if such a name is not otherwise provided.

A logon mode entry determines which entry in the applicable logon mode table is to be used to provide a set of session parameters for the application program if the application program participates as the secondary end of a session (SLU). The name in the DLOGMOD operand must be the name of an entry in a logon mode table. If you do not supply a logon mode table for the application program, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table.

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

For more information, see the description of MODETAB later in this section, and also in *VTAM Customization.*

EAS = n|491
is the approximate number of concurrent sessions this application program will have with other logical units (LU-LU sessions). VTAM uses this operand in a lookup scheme to find the representation of a session between the application program and a logical unit. *n* is a decimal integer from 0 to 32767:

- If *n* is less than 30, VTAM sets up one queue of Function Management Control Blocks (FMCBs).
- If *n* is between 30 and 2000, VTAM sets up 491 queues of FMCBs (VM 404 queues in V3R1).
- If *n* is more than 2000, VTAM sets up 997 queues of FMCBs.

If the actual number of concurrently active sessions with LUs is greater than the number coded in EAS, or if the actual number is greater than 8080, the VTAM mainline path-length is increased.

ENCR = REQD|SEL|OPT|NONE MVS
determines whether this application program has any special requirements for enciphering and deciphering request units.

REQD
means that VTAM must encipher all messages that this application program sends and decipher all messages that the application program receives. ENCR = REQD 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 = REQD, this application program cannot be activated unless the host in which it resides is capable of handling cryptographic sessions.

SEL

means that this application program can choose which messages are to be 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 activated unless the host in which it resides is capable of handling cryptographic sessions.

OPT

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

NONE

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

HAVAIL=YES|NO MVS/XA

determines whether the application can support Extended Recovery Facility (XRF) sessions.

HAVAIL=YES means 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.

HAVAIL=NO means that this application program cannot have an XRF session. HAVAIL=NO is the default.

MAXPVT=0|n|nK|nM MVS VM

is the maximum additional amount of private storage that VTAM can use for session−related control blocks and messages for this application program. The limit is maintained at the task level (not by the individual application program). The task limit is changed when an OPEN or CLOSE ACB is issued. The MAXPVT for any application program is the sum of the MAXPVT specifications of the currently active application programs (that is, those with OPEN ACBs) running under the task.

This number can be expressed as a single number by up to 8 digits or up to 7 digits and M or K. MVS/XA The limit for MAXPVT is approximately 2 billion bytes, which can be expressed as 2048M; MVS/370 VM the limit is approximately 8 megabytes, which can be expressed as 8M.

MAXPVT is ignored if you code it in a VSE or a V3R1 VM system.

0

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

n|nK

n is the number of 1–kilobyte segments that this application program task can use of its private area (address space). *n* is a decimal integer. Its value is rounded up to the next multiple of 4. 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').

nM

n is the maximum amount of private area storage, in 1–megabyte increments, that VTAM can use. *n* is a decimal integer.

MODETAB=logon mode table name

is 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 coded on the MODETAB operand must be the name of a logon mode table created as described in *VTAM Customization*. If you do not supply a logon mode table for the application program, an IBM–supplied logon mode table is used. The user can replace or modify the IBM–supplied logon mode table.

PARSESS=YES|<u>NO</u>

determines whether this application program can have multiple concurrent sessions (parallel sessions) with another application program. If you code YES, the application program can have more than one concurrent LU–LU session with another application program. This other application may be either in the same domain or in another domain. If you code NO or take the default, this application program cannot establish parallel sessions. See *Network Program Products Planning* for the effect of coding YES on the maximum number of minor nodes which can be defined in the host subarea.

PRTCT=password

is a 1– to 8–EBCDIC character password. VTAM compares this password to the one in the application program's access method control block (ACB) when an OPEN macro instruction is issued. It uses this password to verify this application program's authority to run (as the program being defined by this APPL definition statement). If you do not code this operand, no password checking is done.

SONSCIP=YES|<u>NO</u>

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

If you code YES, VTAM terminates the session on behalf of 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 nature of the failure. For a description of these type–codes, see *VTAM Programming*.

If you code NO or take the default, 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.

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=(spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

SRBEXIT=YES|NO MVS
determines whether this application program is authorized to use service request block (SRB) processing in its exit routines. If yes, all the application program's exit routines will be branch entered in SRB mode, supervisor state, key 0. An application must be either APF authorized, key 0–7, or in supervisor state to open an ACB whose corresponding APPL statement includes SRBEXIT=YES.

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

SSCPFM names the system USS table that is to be used when sending a VTAM operator message to the program operator if the message is not defined in a user–defined table. SSCPFM has no effect on the USS table that is used to process VTAM operator commands.

- SSCPFM=USSNOP means that ISTINCNO is the name of the system USS table to be used for VTAM messages sent to this program operator. Note that ISTINCNO is used only for messages that are not defined in a user–defined table.
- SSCPFM=USSPOI means that ISTCFCMM is the name of the system USS table to be used for VTAM messages sent to this program operator. Note that ISTCFCMM is used only for messages that are not defined in a user–defined table.

The default (SSCPFM=USSPOI) is recommended: ISTCFCMM's representation of variable message text works better for CLISTs in NetView or NCCF, and ISTCFCMM guarantees that blanks are not suppressed in messages.

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

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

VPACING=n|0

is the maximum number of normal-flow requests that another LU 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 not from 0 through 63, the maximum of 63 is used. No pacing of requests to the application program is done if VPACING=0 or if you do not code it.

For more information on pacing, refer to "Defining Session Pacing Values" on page 76.

VTAMFRR=YES|NO MVS

determines whether a functional recovery routine (VTAM FRR) will be in effect when control is returned to the application program:

1. After an SRB-authorized path application program interface (API) request

2. When an SRB exit routine is to be scheduled.

If VTAMFRR=YES, then a VTAM functional recovery routine (FRR) will always be in effect when the application program (1) receives control back from an SRB mode API request or (2) receives control in an SRB exit routine. The default is VTAMFRR=NO, which means that VTAM will return control after purging any VTAM FRRs.

Defining Link – Attached Non – SNA Devices to VTAM

Non – SNA start – stop devices must be defined as if they were physical units and logical units. Figure 22 illustrates the operands to use in NCP major nodes and switched major nodes.

A BSC 3270 device attached to a communication controller is supported by VTAM when it is defined in a TERMINAL definition statement with PU = YES on the device's associated GROUP or LINE statement. This allows the device to communicate with VTAM application programs as if it were a physical unit and all terminals attached to it as if they were logical units.

Specify these Operands	In these NCP Generation Macro Inctructions				And these Switched Major Node Definition Statements [1]	
	GROUP	LINE	PU	LU	PU	LU
BATCH=NO	(S)	(S)	(S)	X		
DLOGMOD[2] =logmode table entry name	(O) (S)	(O) (S)	(O) (S)	(O) X	(O) (S)	(O) X
FEATUR2=(operand,operand...) [3]	(O) (S)	(O) (S)	(O) (S)	(O) X	(O) (S)	(O) X
IDNUM=yyyyy,IDBLK=xxx [4]					X	
ISTATUS	(O) (S)	(O) X	(O) X	(O) X	(O) (S)	(O) X
LOCADDR=0				X		
MAXDATA=size [5]	(S)	(S)	X			
MAXLU=1	(S)	(S)	X			
MAXOUT=1	(S)	(S)	X			
MODETAB=logon mode table name [6]	(O) (S)	(O) (S)	(O) (S)	(O) X	(O) (S)	(O) X
PACING=1	(S)	(S)	(S)	X	(S)	X
PASSLIM=1	(S)	(S)	X		X	
PUTYPE=1	(S)	(S)	X		X	
SSCPFM=USSNTO [7]	(S)	(S)	(S)	X	(S)	X
TERM=terminal type [8]			(O) (S)	(O) X	(O) (S)	(O) X
USSTAB=USS definition table name	(O) (S)	(O) (S)	(O) (S)	(O) X	(O) (S)	(O) X
VIRTUAL=YES	X					

Figure 22. Defining non – SNA Start – Stop Devices to VTAM

Legend

(O) operand is optional.
(S) operand can be coded here to take advantage of "sifting".
X operand applies to this definition statement and should be coded here, if not on a higher – level definition statement.

Notes on Figure 22:

1. These definition statements are described in "Defining Switched Major Nodes" on page 152.
2. If you omit DLOGMOD, VTAM uses the first entry of the appropriate logon mode table.
3. These are the features of the non – SNA terminal. VTAM returns this information to an application program that issues an INQUIRE DEVCHAR macro to determine the terminal's features.
4. For multiple terminal access (MTA) devices, IDBLK and IDNUM together define the station ID in the form ID=0100xxxyyyy. Additionally, if the virtual line is defined as an MTA line, you must code an additional PU and LU pair in the VTAM switched major node for each terminal type.

   If the non – SNA device is a TWX terminal that passes an ID sequence when dial – in connection is made, the ID forwarded to VTAM is the ID coded on the NETOLINE macro with the last 20 bits (IDNUM) replaced by the first 20 bits of the TWX ID. There is no dial – out support.
5. The value you choose should reflect system and terminal requirements.
6. If you do not code MODETAB, VTAM uses the IBM – supplied logon mode table named ISTINCLM.
7. USSNTO means that non – SNA devices are supported with the Network Terminal Option program product and these devices use the USS command facilities (LOGON, LOGOFF, AND IBMTEST commands).
8. TERM denotes the type of data stream characteristics (for example, 2740) represented by this logical unit. VTAM returns this value to an application program that issues an INQUIRE DEVCHAR macro to determine the type of data stream supported by the virtual logical unit.

Defining Channel – Attachment Major Nodes

A channel – attachment major node is defined by filing a single VBUILD statement for the major node and separate GROUP, LINE, PU, LU, CLUSTER, and TERMINAL statements for each minor node.

A channel – attachment major node may contain definition statements for:

- Channel – to – channel support
- Channel – attached NCP support (VM Does not apply to V3R1)
- VSE VM Communication adapter support.

You can file these definition statements as one channel – attachment major node or as separate major nodes. Operational considerations such as activation processing overhead or switched network backup capabilities can be used to determine a practical number of major nodes.

A channel – attachment major node begins with a VBUILD statement, but it may contain different types of link groups. Each link group has a unique GROUP 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. The following list shows the structure of minor nodes associated with each type of link group that can be defined in a channel – attachment major node.

- GROUP LNCTL = CTCA (Channel – to – channel support)

  LINE
  PU.

- GROUP LNCTL = NCP (Channel – attached NCP support)

  LINE
  PU.

- VSE VM GROUP LNCTL = SDLC,DIAL = NO (Integrated communication adapter support)

  LINE
  PU
  LU.

- VSE VM GROUP LNCTL = SDLC,DIAL = YES (Integrated communication adapter support)

  LINE
  PU.

- VSE VM GROUP LNCTL = BSC (Integrated communication adapter support)

  LINE
  CLUSTER
  TERMINAL.

	Definition Statements					
Operand	LNCTL = CTCA			LNCTL = NCP		
	GROUP	LINE	PU	GROUP	LINE	PU
ADDRESS		X			X	
CHANCON				S	S	X
DELAY	S	S	X			
ISTATUS	S	X	X	S	X	X
LNCTL	X			X		
MAXBFRU	S	X		S	X	
MAXDATA				S	S	X
PUTYPE	S	S	X	S	S	X
REPLYTO	X					
SPAN	X	X	X	X	X	X

Legend

X = Operand applies to this definition statement and should be coded here, if not in a higher – level definition statement.

S = This is a lower – level operand that can be coded on this definition statement to take advantage of sifting.

Figure 23. Summary of Operands for Channel – Attachment Minor Node Statements

Channel – Attachment Major Node: VBUILD

The VBUILD Statement

Code a VBUILD statement for each set of channel – attachment definition statements.

Name	Definition Statement	Operands
[name]	VBUILD	TYPE = CA [,CONFGDS = name] [,CONFGPW = password]

name
a *name* is optional.

TYPE = CA
means that the VBUILD statement defines a channel – attachment major node to VTAM. This operand is required.

CONFGDS = name
is a 1 – to 8 – character data definition name that identifies the configuration restart data set defined by the user to be used for this major node.

MVS Include a DD statement that has this data definition name in the VTAM start procedure.

Refer to *Network Program Products Planning* for a description of configuration restart.

CONFGPW = password
is the 1 – to 7 – character alphanumeric password, if required, for VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM prompts the VTAM operator for the correct password when VTAM attempts to open the data set.

You may code CONFGPW only if you also code CONFGDS.

VM Both CONFGDS and CONFGPW are ignored if coded on a VM system.

Defining Channel – to – Channel Support

The GROUP Definition Statement

The GROUP definition statement defines the type of links that follow and also carries other operands that "sift down" to following statements that do not explicitly override them. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

This GROUP statement shows the valid operands and defaults for a CTCA line group:

Name	Definition Statement	Operands
name	GROUP	LNCTL = CTCA [,DELAY = time|0|**.100**][1] [,ISTATUS = **ACTIVE**|INACTIVE][2] [,MAXBFRU = ([norm|**10**][,max|**norm**])][3] [,MIH = YES|**NO**] [,PUTYPE = **4**][1] [,REPLYTO = time|**3.0**] [,SPAN = (spanname)]

Notes:

1. *You can code these operands in the GROUP, LINE, or PU statement to take advantage of the sift – down effect, although they are meaningful only for physical units. Their explanations appear in the PU statement description.*

2. *You can code this operand in the GROUP, LINE, or PU statement to take advantage of the sift – down effect, although it is meaningful only for lines and physical units. Its explanation appears in the LINE and PU statement descriptions.*

3. *You can code this operand in either the GROUP or LINE statement to take advantage of the sift – down effect, although it is meaningful only for the channel. Its explanation appears in the LINE statement description.*

name
is the minor node name of the line group represented by this statement. This operand is required.

LNCTL = CTCA
defines the following links as channel – to – channel attachment links. This operand is required.

MIH = YES|NO
causes the link to become inoperative if a Start I/O timeout occurs because the receiving host is in Disabled Compatibility mode. The default is NO.

REPLYTO = time|3.0
tells VTAM how long to wait for completion after starting a channel program. Express *time* in seconds or seconds and tenths of a second. Any value from 0.1 to 25.5 is acceptable. The default of 3.0 seconds should be adequate unless the remote system disables for long intervals. Should this timeout expire during normal operation, a message is written to inform the operator that a timeout has occurred.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Channel – Attachment Major Node: LINE for CTCA

The LINE Definition Statement

Code one LINE definition statement for each channel – to – channel adapter. The LINE statement defines to VTAM the characteristics of its side of the adapter.

Name	Definition Statement	Operands
name	LINE	[ADDRESS = channel unit address] [,DELAY = time|0|**.100**][1] [,ISTATUS = **ACTIVE**|INACTIVE] [,MAXBFRU = ([norm|**10**][,max|**norm**])] [,MIH = YES|**NO**] [,PUTYPE = **4**][1] [,SPAN = (spanname)]

Notes:

1. *You can code these operands in the GROUP, LINE, or PU statement to take advantage of the sift – down effect, although they are meaningful only for physical units. Their explanations appear in the PU statement description. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.*

name
is the minor node name of the link represented by this statement. This operand is required.

ADDRESS = **channel unit address**
is the channel unit address of the channel – to – channel adapter. If the ADDRESS operand is omitted, the VTAM operator must supply the channel unit address by using the U operand of the VARY ACT command.

ISTATUS = **ACTIVE|INACTIVE**
determines whether the line is to be activated after the first activation of the channel – attachment major node and provides a sifting value for the PU statement following this line.

When ISTATUS = ACTIVE is coded or assumed by default, this line is activated as part of the major node activation. If ISTATUS = INACTIVE is coded, this line is left inactive after the first activation of an inactive major node that contains this line. Later activations of an already active major node activate all the lines not previously active (including the ones for which you have coded ISTATUS = INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

MAXBFRU = ([norm|10][,max|norm])

is the number of buffers VTAM will use to receive the data when it starts a normal channel program.

norm

is the number of buffers VTAM will allocate to receive data whenever it starts a normal channel program. The value of *norm* should be somewhat larger than that required for the average data transfer; if *norm* buffers can hold the data, the path through VTAM is shorter than if more than *norm* but less than *max* buffers are required.

The default value is 10 buffers.

max

is the maximum number of buffers VTAM will allocate to receive data for a channel program. The value for *max* must allocate enough buffers to hold the largest path information unit (PIU) that can be received over the channel – to – channel adapter. If you do not code *max*, it defaults to the value used for *norm*.

MIH = YES|NO

causes the link to become inoperative if a Start I/O timeout occurs because the receiving host is in Disabled Compatibility mode. The default is NO.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide*.

Channel – Attachment Major Node: PU for CTCA

The PU Definition Statement

Code one and only one PU definition statement for each LINE statement. (If you are defining a multi – dropped SDLC environment, you can code more than one PU statement for each LINE statement.) VTAM uses the PU statement to define the link station for an adjacent host processor.

Name	Definition Statement	Operands
name	PU	[DELAY = time|0|<u>.100</u>] [,ISTATUS = <u>ACTIVE</u>|INACTIVE] [,PUTYPE = <u>4</u>] [,SPAN = (spanname)]

name
is the minor node name of the physical unit represented by this statement. This operand is required.

DELAY = time|0|.100
is a time interval in seconds. Any value from 0 through 9.999 is acceptable. Three decimal places are allowed. DELAY is the maximum time VTAM should wait after getting low priority (that is, PIU transmission priority 0 or 1) data to send to the other VTAM. This is done in the hope that more data will arrive so that several blocks can be sent together. For most efficient operation, this value should be set as high as possible. For maximum speed across the interface this value should be set to 0. (See the description of the DELAY operand in a channel – to – channel – attached host processor in *VTAM Customization* for more information.)

ISTATUS = ACTIVE|INACTIVE
tells VTAM whether to activate the physical unit automatically when the LINE is activated following the first activation of the channel – attachment major node.

When ISTATUS = ACTIVE (the default), this physical unit is activated automatically when the LINE is activated after the first activation of the major node. When ISTATUS = INACTIVE, this physical unit is left inactive after the first activation of an inactive major node that contains the LINE for this physical unit. Later activations of an already active major node activate all the lines not previously active and their physical units (including the ones for which you coded ISTATUS = INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node or the line.

PUTYPE = 4
is the physical unit type. PUTYPE = 4 is the only valid value and is also the default.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Channel – Attachment Major Node: Considerations for NCP

Defining Channel – Attached NCP Support

Considerations for Defining a Host – to – NCP Channel

You can define a channel attachment between a host and an NCP in a channel – attachment major node. (VM This function is not available in V3R1.) When you plan to do this, you should be aware of other considerations. These considerations include:

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

Figure 24 shows a possible communication management configuration (CMC). In this figure, NCP1 is channel attached to HOST1, but it is in HOST2's domain. NCP1 is a channel – attached cross – domain NCP with respect to HOST1. The same relationship applies between NCP3 and HOST3. All three NCPs are controlled by HOST2, the CMC host. HOST1 and HOST3 can contact their channel – attached NCPs and communicate with resources attached to them without ever activating those NCPs. CMC HOST2 is responsible for activating all three NCPs.

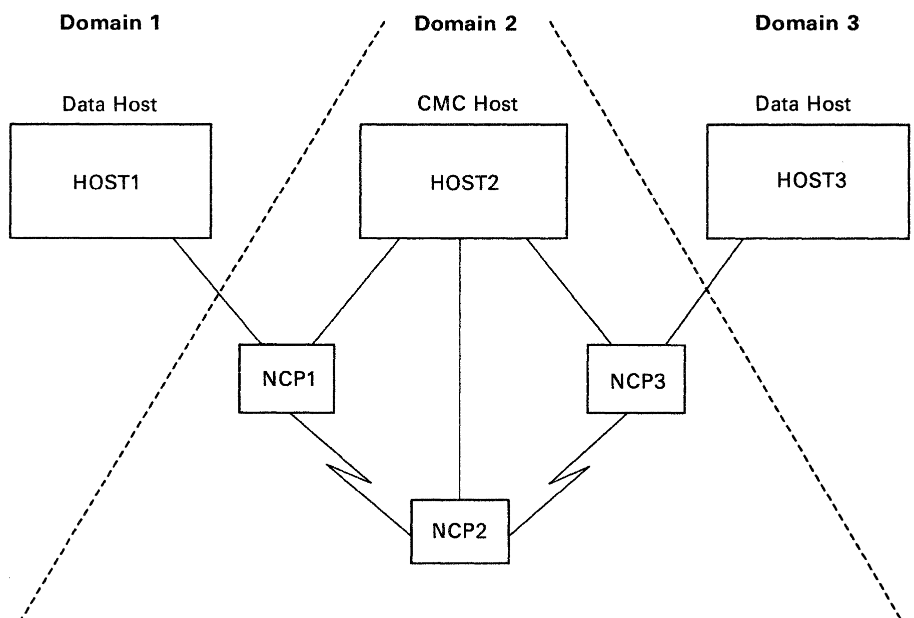


Figure 24. Example of a Communication Management Configuration (CMC)

Channel – Attachment Major Node: Considerations for NCP

Providing for Backup of a CMC Host: The channel connection between HOST2 and NCP1 is defined to HOST2 in a channel – attachment major node. The normal operating activation procedure for a CMC configuration consists of these steps:

1. HOST2 activates its channel – attachment major node.
2. HOST2 activates NCP1.
3. HOST1 activates its channel – attachment major node to contact NCP1.

If for some reason, CMC HOST2 fails and must be backed up by data HOST1, then HOST1 activates NCP1. Activating the NCP establishes an SSCP – PU session between HOST1 and NCP1; HOST1 now becomes the new CMC host. HOST3 can provide backup for HOST2 in much the same way.

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 NCP major node is processed as in VTAM V2 R1 and earlier releases. In this case, 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 contain operands similar to those on the PCCU definition statements. These operands are:

- CHANCON = <u>**COND**</u>|UNCOND

  For a data host, the CHANCON operand on the PCCU definition statement is replaced by the CHANCON operand on the LINE statement in the channel – attachment major node.

- CUADDR = channel device address

  For data hosts, the CUADDR operand on the PCCU definition statement is replaced by the ADDRESS operand on the LINE statement in the channel – attachment major node.

- MAXDATA = size

  For data hosts, the MAXDATA operand on the PCCU definition statement is replaced by the MAXDATA operand on the PU 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 NCP1 is activated, these same operands, if they appear on NCP1's PCCU definition statements are ignored. If you are defining 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.

Channel – Attachment Major Node: Considerations for NCP

The RNAME operand on the PCCU definition statement may 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.

The values defined on the PU statement in the channel – attachment major node are used for the CUADDR, CHANCON, and MAXDATA operands. The PCCU values for these operands are used only when the channel link station is not defined in a channel – attachment major node.

If the access method in the data host has already contacted the NCP, the PCCU operands (CHANCON, CUADDR, and MAXDATA) coded in the NCP major node are not used.

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 will be 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 statement for that major node defines the number of buffers the host can use for receiving data. Figure 24 on page 101 is an example of this. When HOST1 activates its channel – attachment major node, VTAM uses the value of the MAXBFRU operand on the LINE statement. Later, if NCP1 is activated by HOST1, the MAXBFRU value on the HOST definition statement is ignored.

When you define the channel in a channel – attachment major node, VTAM also 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.

Channel – Attachment Major Node: GROUP for NCP

The GROUP Definition Statement

This GROUP definition statement defines channel – to – NCP links. It also carries other operands that "sift down" to following statements that do not explicitly override them. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

When you code LNCTL = NCP, it is not necessary for VTAM to have knowledge of the resources attached to the NCP. If a line is defined in more than one line group, only one group can be active.

Name	Definition Statement	Operands
name	GROUP	LNCTL = NCP [,CHANCON = **COND**|UNCOND][2] [,ISTATUS = **ACTIVE**|INACTIVE][3] [,MAXBFRU = n][1] [,MAXDATA = size|**65535**][2] [,PUTYPE = **4**][2] [,SPAN = (spanname)]

Notes:

1. *You can code this operand in the GROUP statement to take advantage of the sift – down effect, but it applies only to a LINE statement for a channel – to – NCP line. The explanation of the operand appears in the description of the PU statement to which the operand applies.*

2. *You can code these operands in the GROUP statement to take advantage of the sift – down effect, but they apply only to a PU statement for a channel – to – NCP physical unit. The explanation of each operand appears in the description of the PU statement to which the operand applies.*

3. *You can code this operand in the GROUP, LINE, or PU statement to take advantage of the sift – down effect, but it is meaningful only for lines and PUs. Its explanation appears in the LINE and PU statement descriptions.*

name
is the minor node name of the line group represented by this statement. This operand is required.

LNCTL = NCP
defines the following links as channel – to – NCP links. This operand is required.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

The LINE Definition Statement

Code one LINE definition statement for each channel – to – NCP link. The LINE statement defines to VTAM:

- The name for the link.
- The channel unit address for the link.
- Certain procedural options to be used for this line.

Name	Definition Statement	Operands
name	LINE	MAXBFRU = n [,ADDRESS = channel unit address] [,CHANCON = **COND**|UNCOND][1] [,ISTATUS = **ACTIVE**|INACTIVE][2] [,MAXDATA = n|**65535**][1] [,PUTYPE = 4][1] [,SPAN = (spanname)]

Notes:

1. *You can code these operands in the LINE statement to take advantage of the sift – down effect, but they apply only to a PU statement for a channel – to – NCP physical unit. The explanation of each operand appears in the description of the PU statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.*

2. *You can code this operand in the LINE or PU statement, and it is meaningful for both lines and PUs. Its explanation appears in the LINE and PU statement descriptions. However, it can also be coded on the GROUP statement to take advantage of the sift – down effect.*

name
is the 1 – to 8 – character minor node name of the link represented by this statement. This name is required.

Note: This channel link definition may 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 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 SDLC and channel link names, see *Network Program Products Planning.*

MAXBFRU = n
defines the number of buffers to be set aside by the host for receiving data from the NCP. This operand has the same meaning as the MAXBFRU operand on the HOST definition statement in the NCP major node. The minimum value for MAXBFRU is 1; the maximum value is 255.

ADDRESS = channel unit address
is the channel unit address of the channel – attached NCP. If the ADDRESS operand is omitted, the VTAM operator must supply the channel unit address by using the U operand of the VARY ACT command. (See *VTAM Operation* for details on using the U operand.)

ISTATUS = ACTIVE|INACTIVE
determines whether the line is to be activated after the first activation of the channel – attachment major node. It provides a sifting value for the PU statement that follows the LINE statement.

When ISTATUS = ACTIVE is coded or assumed by default, this line is activated as part of the first activation of the channel – attachment major node. If ISTATUS = INACTIVE is coded, this line is left inactive after the first activation of an inactive major node that contains this line. Later activations of an already active major node activate all the lines not previously active (including the ones for which you coded ISTATUS = INACTIVE). Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

The PU Definition Statement

Code one PU definition statement for each LINE statement.

Name	Definition Statement	Operands
name	PU	[CHANCON = **COND**|UNCOND] [,ISTATUS = **ACTIVE**|INACTIVE][1] [,MAXDATA = n|**65535**] [,PUTYPE = **4**] [,SPAN = (spanname)]

Notes:

1. *You can code this operand in the LINE or PU statement, where it directly applies, or on the GROUP statement, where it contains a value to be sifted to lower – level LINE and PU statements. Its explanation appears in the LINE and PU statement descriptions. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.*

name
is the 1 – to 8 – character minor node name of the physical unit represented by this statement. This name is required.

CHANCON = COND|UNCOND
tells VTAM whether to send a conditional or an unconditional channel Contact request to the NCP when the host wishes to activate an NCP over a channel.

If you omit CHANCON, or if CHANCON = COND, VTAM will allow 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). Such a case results if duplicate subareas are assigned to more than one host (attached to more than one channel interface.) Two hosts may have been accidentally assigned the same number or they may have been assigned duplicate numbers to allow mutual backup capabilities.

If you code CHANCON = UNCOND, VTAM will set an indicator in the channel Contact request that will force 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 with the same subarea number is channel – attached to the NCP, the NCP will break contact with the host on the other channel.

ISTATUS = ACTIVE|INACTIVE
determines whether the physical unit is to be activated after the first activation of the channel – attachment major node. When ISTATUS = INACTIVE, this physical unit is left inactive after the first activation of an inactive major node that contains the LINE for this physical unit. Later activations of an already active major node activate all the lines not previously active and their physical units (including the ones for which you coded ISTATUS = INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

MAXDATA = n|65535

is the maximum amount of data in bytes—including the transmission header (TH) and the request/response header (RH)—that the NCP can receive in one segment of a path information unit (PIU). The MAXDATA size should, if possible, be equal to the size of the largest PIU the network will handle, but should not exceed the size of MVS VM the IOBUF buffer pool; VSE the VFBUF buffer pool. It also should not exceed the product of the MAXBFRU and UNITSZ values for the NCP. The default is the maximum size of 65535 bytes.

This operand applies only to the PCCU definition statement for a channel – attached NCP. However, the MAXDATA value should also not exceed the capacities of any link – attached NCPs or of any cross – domain NCPs. An SNA path error can result if the MAXDATA value exceeds the capacities of the NCPs along the route of a PIU.

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 statement.

PUTYPE = 4

is the physical unit type. PUTYPE = 4 is the only valid value and is also the default.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Channel – Attachment Major Node: GROUP for SDLC Nonswitched

VSE VM Defining SDLC and BSC Line Support

VSE VM The GROUP Statement for an SDLC Nonswitched Line Group

This GROUP statement defines an SDLC nonswitched line group. It also carries other operands that “sift down” unless you code PU and LU statements to override them explicitly.

If a line is defined in more than one line group, only one of the groups can be active at a time.

Name	Definition Statement	Operands
name	GROUP	LNCTL=SDLC [,ACTIVTO=t][1] [,DIAL=NO] [,DISCNT=([YES|**NO**] [,**F**|NF])][2] [,DLOGMOD=default logmode entry name][3] [,ISTATUS=**ACTIVE**|INACTIVE][4] [,LOGAPPL=application program name][3] [,LOGTAB=interpret table name][3] [,MAXBFRU=([norm|**1**|**2**] [,max|**2**|**8**])][2] [,MAXDATA=size|**261**|**265**][2] [,MAXOUT=n|**1**][2] [,MODETAB=logon mode table name][3] [,PACING=n|0|**1**][3] [,PASSLIM=n|**maxout**][2] [,PAUSE=t|**0.1**][1] [,PUTYPE=1|**2**|4|5][2] [,REPLYTO=n|**1**][1] [,RETRIES=n|**7**][2] [,SERVLIM=n|**4**][1] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS][3] [,USSTAB=USS definition table][3] [,VPACING=n|0|**2**][3]

Notes:

1. *Applies to a LINE definition statement for an SDLC nonswitched line.*

2. *Applies to a PU definition statement for a physical unit on an SDLC nonswitched line.*

3. *Applies to an LU definition statement.*

4. *You can code this operand in either the GROUP, LINE, PU, or LU statement. It is meaningful for lines, physical units, and logical units. Its explanation appears in the LINE, PU, and LU statement descriptions.*

You can code these operands on the GROUP definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See “Sift – Down Effect in VTAM Definition Statements” on page 71 for more information.

name
provides the minor node name for the line group. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character.

LNCTL = SDLC
means that the group contains synchronous data link control (SDLC) lines. A line group may contain lines of only one type.

DIAL = NO
means that the lines in the group do not require switched line control protocols.

Note: SDLC nonswitched lines that have switched network backup capability should be defined as DIAL = NO.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The LINE Statement for an SDLC Nonswitched Line

Code one LINE statement for each SDLC nonswitched line. The LINE statement defines to VTAM:

- The name of the line
- The channel unit address for the line
- Certain procedural options to be used for this line.

Name	Definition Statement	Operands
name	LINE	[ADDRESS = channel unit address|**030**] [,ACTIVTO = t] [,DISCNT = ([YES|**NO**] [,**F**|NF])][1] [,DLOGMOD = default logmode entry name][2] [,ISTATUS = **ACTIVE**|INACTIVE][3] [,LOGAPPL = application program name][2] [,LOGTAB = interpret table name][2] [,MAXBFRU = ([norm|**1**|**2**] [,max|**2**|**8**])] [1] [,MAXDATA = size|**261**|**265**][1] [,MAXOUT = n|**1**][1] [,MODETAB = logon mode table name][2] [,PACING = n|0|**1**][2] [,PASSLIM = n|**maxout**][1] [,PAUSE = t|**0.1**] [,PUTYPE = 1|**2**|4|5][1] [,REPLYTO = n|**1**] [,RETRIES = n|**7**][1] [,SERVLIM = n|**4**] [,SPAN = (spanname)] [,SSCPFM = **FSS**|USSSCS][2] [,USSTAB = USS definition table][2] [,VPACING = n|0|**2**][2]

Notes:

1. *Applies to a PU definition statement for a physical unit on an SDLC nonswitched line.*

2. *Applies to an LU definition statement.*

3. *You can code this operand in either the GROUP, LINE, PU, or LU statement. It is meaningful for lines, physical units, and logical units. Its explanation appears in the LINE, PU, and LU statement descriptions.*

You can code these operands in the LINE definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

name
provides the minor node name for the communication line. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character.

ADDRESS=cua|030

is the channel unit address for the SDLC nonswitched line represented by this LINE statement. The defined address can be overridden by the U= operand of the VARY ACT command.

ACTIVTO=t

is the interval that the communication adapter will wait without detecting an SDLC frame from another domain's NCP (PUTYPE=4) or from another VTAM with a communication adapter (PUTYPE=5). After this interval, the communication adapter will signal a time–out error to VTAM (SDLC nonproductive receive time–out).

t

is the interval in seconds. The valid range is 1 through 255.

ACTIVTO is defaulted to 60 seconds unless you code REPLYTO. When you code REPLYTO, ACTIVTO is defaulted to a value that is equal to ten (10) times the REPLYTO value.

ISTATUS=ACTIVE|INACTIVE

determines whether this line is activated when the channel–attachment major node to which it belongs is activated and provides a sifting value for the PU and LU statements following this line. This can be overridden by the SCOPE operand on the VARY ACT command when the major node is activated.

MAXBFRU=([norm|1|2] [,max|2|8])

is the number of buffers VTAM will use whenever it starts a normal channel program.

norm|1|2

is the number of buffers VTAM will use whenever it reads 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 less than *max,* buffers are required.

The default for *norm* is 1 for type 1 and 2 physical units and 2 for type 4 and 5 physical units.

max|2|8

is the maximum number of buffers VTAM will use 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,* it defaults to 2 for type 1 and 2 physical units and to 8 for type 4 and 5 physical units.

PAUSE = t|0.1
is the interval that the communication adapter will wait after completing a normal (data) poll cycle before signaling completion if all stations respond negatively (no data to send).

t
is the interval in seconds and tenths of seconds. The valid range is 0 through 25.5 seconds.

Note: This operand is not valid for a line associated with a type 4 physical unit.

REPLYTO = t|1.0
is the reply time – out value for the line when operating as the primary station. If, at the expiration of this interval, the communication adapter has not received a response to poll from the secondary station, 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 defined by RETRIES.

t
is the time – out value, expressed in seconds or seconds and tenths of seconds. The valid range is 0.1 to 25.5 seconds.

SERVLIM = n|4
is the ratio of data poll cycles to contact poll cycles, that is, how many times each active physical unit on the line is asked to send information before attempting to contact additional physical units.

n
n is an integer from 0 through 255. For guidance as to appropriate values, consult the relevant publication for that device.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The PU Statement for a Physical Unit on an SDLC Nonswitched Line

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

The PU statement defines:

- The resource name for the physical unit
- The SDLC station address of the physical unit
- The physical unit type (1, 2, 4, or 5)
- The maximum size of a PIU or PIU segment that VTAM will send to the physical unit
- The maximum number of path information units (PIUs) or PIU segments VTAM will send to the physical unit before requesting a response from the unit
- The maximum number of consecutive path information units (PIUs) or PIU segments VTAM will send to the physical unit before servicing other physical units on the link
- The number of error recovery attempts VTAM will make when transmission errors occur
- The subarea address of the physical unit (types 4 and 5 only).

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

Channel – Attachment Major Node: PU for SDLC Nonswitched

Name	Definition Statement	Operands
name	PU	ADDR = char [,DISCNT = ([YES|**NO**] [,**F**|NF])] [,DLOGMOD = default logmode entry name][1] [,ISTATUS = **ACTIVE**|INACTIVE][2] [,LOGAPPL = application program name][1] [,LOGTAB = interpret table name][1] [,MAXDATA = size|**261**|**265**] [,MAXOUT = n|**1**] [,MODETAB = logon mode table name][1] [,PACING = n|0|**1**][1] [,PASSLIM = n|**maxout**] [,PUTYPE = 1|**2**|4|5] [,RETRIES = n|**7**] [,SPAN = (spanname)] [,SSCPFM = **FSS**|USSSCS][1] [,SUBAREA = n] [,TADDR = chars|**C1**] [,USSTAB = USS definition table][1] [,VPACING = n|0|**2**][1]

Notes:

1. *Applies to LU definition statement.*

2. *You can code this operand in either the GROUP, LINE, PU, or LU statement. It is meaningful for lines, physical units, and logical units. Its explanation appears in the LINE, PU, and LU statement descriptions.*

You can code these operands in the PU definition statement. The explanation of each operand appears in the description of the LU definition statement. See “Sift – Down Effect in VTAM Definition Statements” on page 71 for more information.

name

is the minor node name of the physical unit represented by this statement. This operand is required.

ADDR = char

is the physical unit's SDLC address.

char

is the 2 – digit hexadecimal SDLC station address. This address must be unique for each physical unit on the same line. The valid range is 01 through FE. Consult the relevant device publication for guidance in assigning this value.

Example: If the station address of a 3601 is ‘C1’, code ADDR = C1 on the PU definition statement.

Note: This operand is required for PU types 1 and 2. If you code PUTYPE = 5, ADDR defaults to C1; you do not have to give any other value for a PU type 5 because the default (C1) will always work.

DISCNT = ([YES|NO] [,F|NF])

YES|NO
tells VTAM whether to terminate its SSCP – PU and SSCP – LU sessions when the last associated LU – LU session (application program – to – logical unit session) is terminated.

YES
tells VTAM to automatically terminate the SSCP – PU and SSCP – LU sessions as soon as the last logical unit – to – application program (LU – LU) session is terminated. If any of the logical units request their own session termination, VTAM ignores the HOLD part of a character – coded logoff or the LAST – NOTLAST part of a field – formatted Terminate Self request. VTAM also rejects any attempt made by the physical unit to terminate its own SSCP session with a Request Discontact request. Before the physical unit and its logical units can be used again, they must be reactivated by the domain operator.

NO
tells VTAM to terminate the SSCP – PU and SSCP – LU sessions when one of the following conditions is met:

- VTAM receives a Request Discontact Normal request from the physical unit. VTAM allows any LU – LU sessions associated with that physical unit to terminate normally.
- VTAM receives a Request Discontact Immediate request from the physical unit. VTAM immediately terminates any existing LU – LU sessions associated with that physical unit. This request overrides any previous Request Discontact Normal request from the physical unit.
- All LU – LU sessions have been terminated as a result of a character – coded logoff with HOLD = NO, a Terminate Self request with LAST specified, or a VARY INACT command. If any LU – LU sessions were terminated by any other means (for example, by an application program CLSDST macro instruction), the SSCP – PU and SSCP – LU sessions are not terminated.

F|NF
determines whether VTAM is to indicate "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 take the default, "final – use" status is indicated and the physical connection may be broken. 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.

ISTATUS = ACTIVE|INACTIVE

provides a "sifting" value for the LU statements following this PU statement. If ISTATUS = ACTIVE, a physical unit is activated automatically when the line to which it is attached is activated. Note that you can override this with the SCOPE operand on the VARY ACT command when the major node or the line is activated.

MAXDATA = size|261 (for PU type 1)|265 (for PU type 2)

is the maximum number of bytes that the physical unit can receive in one path information unit (PIU) or PIU segment. A PIU includes data and both the transmission header and the request/response header.

size is an integer from of 5 through 65535.

To determine the maximum PIU (or PIU segment) size that the physical unit can receive, consult the publications for the specific type of SDLC station represented by this PU statement. This value varies with the device; for example, the 3276 requires MAXDATA = 262.

The maximum amount of user data that VTAM sends to the physical unit in one PIU is the value of MAXDATA minus 5 bytes (for a PU type 1) or minus 9 bytes (for a PU type 2). (These subtracted values represent the lengths of the request/response header [always 3 bytes] and the transmission header [either 2 bytes for a PU type 1 or 6 bytes for a PU type 2].)

MAXOUT = n|1

is the maximum number of path information units (PIUs) (or PIU segments if the session parameters allow segmenting of data) that VTAM will send to the physical unit represented by this statement before requesting a response.

n is an integer from 1 through 7.

PASSLIM = n|maxout

is the maximum number of consecutive path information units (PIUs) or PIU segments VTAM will send at one time to the physical unit represented by this PU statement.

n is an integer from 1 through 254.

If you omit PASSLIM or code it incorrectly, the value of MAXOUT is used.

PUTYPE = 1|2|4|5

is the physical unit type constituting the SDLC station represented by this PU definition statement.

1

an SNA terminal.

2
: an SNA cluster controller.

4
: another domain's NCP in a communication controller.

5
: another domain's VTAM and a communication adapter.

VTAM allows PUTYPE = 1 and PUTYPE = 2 stations to be intermixed on the same multipoint line. PUTYPE = 4 and PUTYPE = 5 stations are supported only on point – to – point (single station) lines.

RETRIES = n|7
is the number of times VTAM tries to recover from errors that occur during transmission to or from the physical unit represented by this PU statement.

n is an integer from 0 through 255.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

SUBAREA = n
is the subarea address assigned to another domain's NCP (PUTYPE = 4) or to another domain's VTAM (PUTYPE = 5), whichever is represented by this PU statement.

n is an integer from 1 through the MAXSUBA value. (See "The MAXSUBA Start Option" on page 234 for a description of the MAXSUBA start option.) This value must be the same as the subarea address defined for this physical unit in other domains of the network.

VTAM uses the subarea value to determine the primary of two type 5 physical units; the one with the highest subarea is always primary. (In a connection through a communication adapter, VTAM is always secondary to an NCP, a PU type 4.)

TADDR = char|C1
is the SDLC station address to use when VTAM is acting as a secondary station to a PU type 4 (an NCP). This operand is only valid on a PU definition statement for a type 4 physical unit.

char
is the 2 – digit hexadecimal SDLC station address.

VSE VM The LU Statement for a Logical Unit on an SDLC Nonswitched Line

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

The LU statement defines:

- The resource name for the logical unit
- The local address for the logical unit
- Use of the pacing option.

You must have a separate LU definition statement for each logical unit associated with a PU type 1 or 2 on a nonswitched line. LU statements immediately follow the PU statement for the physical unit with which they are associated.

Name	Definition Statement	Operands
name	LU	LOCADDR = n [,DLOGMOD = default logmode entry name] [,ISTATUS = **ACTIVE**|INACTIVE][1] [,LOGAPPL = application program name] [,LOGTAB = interpret table name] [,MODETAB = logon mode table name] [,PACING = n|0|**1**] [,SPAN = (spanname)] [,SSCPFM = **FSS**|USSSCS] [,USSTAB = USS definition table name] [,VPACING = n|0|**2**]

Notes:

1. *You can code this operand in either the GROUP, LINE, PU, or LU statement. It is meaningful for lines, physical units, and logical units. Its explanation appears in the LINE, PU, and LU statement descriptions.*

name
is the minor node name of the logical unit. This operand is required.

LOCADDR = n
is the local address (a decimal integer, without leading zeros) of the logical unit. The valid range of addresses for logical units associated with a PU type 1 is 0 through 63; for logical units associated with a PU type 2, the valid range is 1 through 255.

Note: The value of LOCADDR depends upon 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 between 2 and 33.

DLOGMOD = default logon mode table entry name
is the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default.

ISTATUS = ACTIVE|INACTIVE
determines whether this logical unit is to be activated when the physical unit is first activated. Note that this can be overridden by the SCOPE operand on the VARY ACT command when the major node is activated.

LOGAPPL = application program name
identifies an application program to which this logical unit is to be automatically logged on when it is activated. The name must correspond to the network unique name assigned to the application program by an APPL definition statement.

If you do not code LOGAPPL, logons are initiated through USS facilities by the terminal operator, an application program, or the domain operator.

In VTAM V3R1.1, you can cancel this automatic logon definition at any time with the VARY NOLOGON command.

For more information on establishing sessions, see *Network Program Products Planning.*

LOGTAB = interpret table name
is the name of an interpret table to be used by VTAM when processing logons originating from the logical unit. See *VTAM Customization.*

MODETAB = logon mode table
is the name of a logon mode table to be used for the logical unit. If this operand is omitted, the IBM – supplied logon mode table is used for the logical unit.

PACING = n|0|1
determines how the flow of data is to be paced from VTAM (the boundary node) to the secondary logical unit. For more information on pacing, see "Defining Session Pacing Values" on page 76.

n

determines how many normal – flow requests VTAM is to send for a given logical unit – logical unit (LU – LU) session before waiting for a pacing response. No further normal – flow requests can be sent to the logical unit until the logical unit is ready to receive more requests.

n is a decimal integer from 1 through 255.

0

means that no pacing is to be performed for sessions with the logical unit.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide*.

SSCPFM = FSS|USSSCS

determines whether the logical unit uses formatted commands (FSS) or character – coded commands (USSSCS) when communicating with the SSCP.

See the publications for each individual device to determine whether formatted or character – coded commands are supported by that device.

USSTAB = USS definition table name

is the name of a USS definition table to be used for the logical unit. USS definition tables are described in *VTAM Customization*. If this operand is omitted, the IBM – supplied USS definition table is used for the logical unit when character – coded requests are received.

VPACING = n|0|2

Determines how the flow of data is to be paced between the primary logical unit and VTAM's boundary function (2 – stage pacing). For more information on pacing, see "Defining Session Pacing Values" on page 76.

n

determines how many normal – flow requests that the primary logical unit is to send for a given logical unit – logical unit (LU – LU) session before waiting for a pacing response. No further normal – flow requests can be sent to the logical unit until VTAM's boundary function is ready to receive more requests.

n is a decimal integer from 1 through 63.

0

means that no pacing is to be performed for session with the logical unit; that is, VTAM will not wait for a pacing response.

VSE VM The GROUP Statement for an SDLC Switched Line Group

This GROUP statement defines an SDLC switched line group; it also carries other operands that "sift down" to following statements that do not explicitly override them.

If a line is defined in more than one line group, only one of the groups can be active.

Name	Definition Statement	Operands
name	GROUP	LNCTL=SDLC ,DIAL=YES [,ACTIVTO=t][1] [,ANSWER=**ON**|OFF][1] [,AUTODL=**YES**|NO][1] VSE [,CALL=**IN**|OUT|INOUT][1] [,DIALNO=telephone number] VSE [,ISTATUS=**ACTIVE**|INACTIVE][2] [,MAXBFRU=([norm|**1**|**2**] [,max|**2**|**8**])][1] [,MAXLU=n|**2**][3] [,PAUSE=t|**0.1**][1] [,REPLYTO=t|**1**][1] [,RETRIES=n|**7**][1] [,RETRYTO=t|**12**][1] VSE [,SERVLIM=n|**4**][1] [,SHOLD=**NO**|(free,npoll)] VSE [,SPAN=spanname] [,X21SW=YES|**NO**]

Notes:

1. *Applies to a LINE definition statement for an SDLC switched line.*

2. *You can code this operand in either the GROUP, LINE, or PU statement. It is meaningful for lines and physical units. Its explanation appears in the LINE and PU statement descriptions.*

3. *Applies to a PU definition statement for a PU on an SDLC switched line.*

You can code these operands in the GROUP definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

name
: provides the minor node name for the line group. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character.

LNCTL=SDLC
: means that the group contains synchronous data link control (SDLC) lines. A line group may contain lines of only one type.

DIAL = YES

means that the lines in the group require switched line control protocols.

DIALNO = telephone number VSE

is the telephone number (in EBCDIC) used to initiate a connection with a physical unit over a switched link. You can insert special characters in the number according to the following rules:

- You can insert as many vertical bars ("|"; hex FA) as you need to denote dialing pauses.
- For separator characters, use underscores ("_"; hex 6D) or vertical bars ("|"; hex FA).
- For end – of – number characters, use asterisks ("*"; hex 5C), percent signs ("%"; hex 6C), or at – signs ("@"; hex 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 *NCP – SSP Resource Definition Reference.*

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 will supply 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 (hex 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.

Instead of coding:

```
P1 PATH DIALNO=5551235  FULL DIAL ADDRESS FOR GROUP G1
```

you would code:

```
P1 PATH DIALNO=.27      ABBREVIATED DIAL ADDRESS FOR GROUP G1
```

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.

DIALNO is required if SHOLD is coded.

For more information on X.21 switched line features, see *General Information: X.21 Interface Features.*

SHOLD = NO|(free,npoll) VSE

tells VTAM whether this group is an X.21 Short Hold Mode/Multiple Port Sharing (SHM/MPS) group. (X.21 Short Hold Mode/Multiple Port Sharing is a feature available with VTAM V3R1.)

SHOLD = NO is the default and means this group is not an SHM/MPS group.

SHOLD = *(free,npoll)* means this group is an SHM/MPS group and defines default values for the lines in the group. These values mean:

free is the free period, in seconds, after dial completion. It is a decimal number between 0.1 and 3276.7.

npoll is the non – productive polling count. It is a decimal number between 1 and 32,767.

If you code SHOLD = *(free,npoll)*, you must also code X21SW = YES and AUTODL = YES. You may allow AUTODL to default to YES.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

X21SW = YES|NO

Tells VTAM whether the lines in the group are X.21 switched lines. If X21SW = YES, every line in the group is an X.21 switched line. You must set this operand to YES if X.21 Short Hold Mode is activated. (X.21 Short Hold Mode/Multiple Port Sharing is a feature available with VTAM V3R1.) The default is *NO.*

VSE VM The LINE Statement for an SDLC Switched Line

Code one LINE statement for each SDLC switched line. The LINE statement defines to VTAM:

- The name of the line
- The channel unit address of the line
- Certain procedural options to be used for this line.

Name	Definition Statement	Operands
name	LINE	[ACTIVTO=t]
		[,ADDRESS=channel unit address|**030**]
		[,ANSWER=**ON**|OFF]
		[,AUTO=address]
		[,AUTODL=**YES**|NO] VSE
		[,CALL=**IN**|OUT|INOUT]
		[,ISTATUS=**ACTIVE**|INACTIVE][2]
		[,MAXBFRU=([norm|**1**] [,max|**2**])]
		[,MAXLU=n|**2**][1]
		[,PAUSE=t|**0.1**]
		[,REPLYTO=t|**1**]
		[,RETRIES=n|**7**]
		[,RETRYTO=t|**12**] VSE
		[,SERVLIM=n|**4**]
		[,SPAN=spanname]

Notes:

1. *Applies to a PU definition statement for a physical unit on an SDLC switched line.*

2. *You can code this operand in either the GROUP, LINE, or PU statement. It is meaningful for lines and physical units. Its explanation appears in the LINE and PU statement descriptions.*

You can code these operands in the LINE definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

name
provides the minor node name for the communication line. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character.

ACTIVTO = t

determines how long the communication adapter will wait without detecting an SDLC frame from another domain's NCP (PUTYPE = 4) or from another VTAM with a communication adapter (PUTYPE = 5). After this interval, the communication adapter will signal a time – out error to VTAM (SDLC nonproductive receive time – out).

t

is the interval in seconds. The valid range is 1 through 255.

ACTIVTO is defaulted to 60 seconds unless you code REPLYTO. When you code REPLYTO, ACTIVTO is defaulted to a value that is equal to ten (10) times the REPLYTO value.

ADDRESS = channel unit address|030

is the channel unit address for the SDLC switched line represented by this LINE statement. The defined address can be overridden by the U= operand of the VARY ACT command.

ANSWER = ON|OFF

determines whether physical units can dial in to VTAM. If ANSWER = ON (the default), physical units can dial in to VTAM when the line is activated. If ANSWER = OFF, the physical units cannot dial in to VTAM, regardless of the active – inactive status of the line. The answer state of the line can be altered after activation by a VARY ANS command.

AUTO = address

is the presence of an automatic calling unit (autocall unit) for this line. The address of the autocall unit is the same as the address of the line. The communication adapter can support up to two autocall units, which may be associated with any of the line addresses. Support for up to two autocall units is an optional feature of the communications adapter.

AUTODL = YES|NO VSE

tells VTAM whether an X.21 switched interface with the direct call facility will accept 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 has been defined with dial digits.

This operand must be set to YES if you coded SHOLD = (*free,npoll*) on the GROUP statement for this line.

CALL = IN|OUT|INOUT
determines whether stations, or VTAM, or both, can initiate calls over the line represented by this LINE statement.

If the line is to be used only for incoming calls (the stations call VTAM), code CALL = IN, which is the default. If the line is to be used only for outgoing calls (VTAM calls the physical unit), call CALL = OUT. If the line is to be used for both incoming and outgoing calls, code CALL = INOUT.

If you code CALL = OUT or you code CALL = INOUT without coding an AUTO operand, outgoing calls are dialed by the operator as directed by a VTAM console message. If X.21 Short Hold Mode is activated, you must code CALL = INOUT. (X.21 Short Hold Mode/Multiple Port Sharing is a feature available with VTAM V3R1.)

ISTATUS = ACTIVE|INACTIVE
determines whether the line is to be activated after the first activation of the channel attachment major node and provides a sifting value for the PU statements following this line. Note that this can be overridden by the SCOPE operand on the VARY ACT command when the major node is activated.

MAXBFRU = ([norm|1] [,max|2])
is the number of buffers VTAM will use whenever it starts a normal channel program.

norm|1
is the number of buffers VTAM will use whenever it reads 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 less than *max,* buffers are required.

The default for *norm* is 1.

max|2
is the maximum number of buffers VTAM will use 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*, it defaults to 2.

PAUSE = t|0.1
determines how long the communication adapter will wait after completing a normal (data) poll cycle before signalling completion if all stations respond negatively (no data to send).

t
is the interval in seconds and tenths of seconds. The valid range is 0 through 25.5 seconds.

Note: This operand is not valid for a line associated with a type 4 physical unit.

REPLYTO = t|1.0
is the reply time – out value for the line when operating as the primary station. If, at the expiration of this interval, the communication adapter has not received a response to poll from the secondary station, it notifies VTAM that a timeout error has occurred (idle detect time – out). VTAM retries the poll, select, or text – write up to the limit defined by RETRIES.

t

is the timeout value, expressed in seconds or in seconds and tenths of seconds. The valid range is 0.1 to 25.5 seconds.

RETRIES = n|7
determines how many times VTAM tries to recover from errors that occur during transmission on this line. *n* is an integer from 0 through 255.

RETRYTO = t|12 VSE
is the time interval that the communication adapter will wait, after receiving a retriable Call Process Signal, before it presents the signal to VTAM. *t* is expressed in seconds and can be 0, 1, 3, 6, 12, 18, or 24. The default is 12 seconds.

This operand is valid only if you coded X21SW = YES on the GROUP statement for this line.

SERVLIM = n|4
is the ratio of data poll cycles to contact poll cycles; that is, how many times each active physical unit on the line is asked to send information before attempting to contact additional physical units.

n

n is an integer from 0 through 255. For guidance as to appropriate values, consult the relevant publication for that device.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The PU Statement for a Physical Unit on an SDLC Switched Line

Code one PU statement for each physical unit with which VTAM will communicate over this switched SDLC link.

The PU statement defines:

- The resource name of the physical unit
- The maximum number of logical units to be associated with the physical unit.

Name	Definition Statement	Operands
name	PU	[ISTATUS = **ACTIVE**|INACTIVE][1] [,MAXLU = n|**2**] [,SPAN = (spanname)]

Notes:

1. *You can code this operand in either the GROUP, LINE, or PU statement. It is meaningful for lines and physical units. Its explanation appears in the LINE and PU statement descriptions.*

name
is the minor node name of the physical unit represented by this statement. This operand is required.

ISTATUS = ACTIVE|INACTIVE
provides a "sifting" value for other minor nodes in this group. If ISTATUS = ACTIVE, a group is always activated when the channel–attachment major node to which it belongs is activated. Note that you can override this with the SCOPE operand on the VARY ACT command when the major node is activated.

MAXLU = n|2
is, for a physical unit on a switched line (DIAL = YES), the maximum number of logical units associated with the physical unit that can communicate with VTAM.

n is an integer from 1 through 255.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The GROUP Statement for a BSC Line Group

This GROUP statement defines a BSC line group. It also carries other operands that "sift down" to following statements that do not explicitly override them.

If a line is defined in more than one line group, only one of the groups can be active.

Name	Definition Statement	Operands
name	GROUP	LNCTL = BSC [,CUTYPE = 3271|3275][2] [,DLOGMOD = default logmode entry name][3] [,FEATUR2 = ([MODEL1|MODEL2] [,PRINTR|NOPRINTR])][3] [,ISTATUS = ACTIVE|INACTIVE][4] [,LOGAPPL = application program name][3] [,LOGTAB = interpret table name][3] [,MODETAB = logon mode table name][3] [,RETRIES = n|7][1] [,SERVLIM = n|4][1] [,SPAN = (spanname)] [,TERM = 3275|3277|3284|3286][3] [,USSTAB = USS definition table][3]

Notes:

1. *Applies to a LINE definition statement for a BSC line.*
2. *Applies to a CLUSTER definition statement.*
3. *Applies to a TERMINAL definition statement.*
4. *You can code this operand in either the GROUP, LINE, CLUSTER, or TERMINAL statement. It is meaningful for lines, clusters, and terminals. Its explanation appears in the LINE, CLUSTER, and TERMINAL statement descriptions.*

You can code these operands in the GROUP definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

name
provides the minor node name for the line group. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character. This operand is required.

LNCTL = BSC
means that the group contains binary synchronous communication (BSC) lines. A line group may contain lines of only one type.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The LINE Statement for a BSC Nonswitched Line

Code one LINE statement for each BSC nonswitched line. The LINE statement defines to VTAM:

- The name of the line
- The channel unit address of the line
- Certain procedural options to be used for this line.

Name	Definition Statement	Operands
name	LINE	[ADDRESS = channel unit address|**030**] [,CUTYPE = 3271|3275][1] [,DLOGMOD = default logmode entry name][2] [,FEATUR2 = ([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])][2] [,ISTATUS = **ACTIVE**|INACTIVE][3] [,LOGAPPL = application program name][2] [,LOGTAB = interpret table name][2] [,MODETAB = logon mode table name][2] [,RETRIES = n|7] [,SERVLIM = n|4] [,SPAN = (spanname)] [,TERM = 3275|3277|3284|3286][2] [,USSTAB = USS definition table][2]

Notes:

1. *Applies to a CLUSTER definition statement.*
2. *Applies to a TERMINAL definition statement.*
3. *You can code this operand in either the GROUP, LINE, CLUSTER, or TERMINAL statement. It is meaningful for lines, clusters, and terminals. Its explanation appears in the LINE, CLUSTER, and TERMINAL statement descriptions.*

You can code these operands in the LINE definition statement. The explanation of each operand appears in the description of the definition statement to which the operand applies. See "Sift – Down Effect in VTAM Definition Statements" on page 71 for more information.

name
provides the minor node name for the communication line. It contains 1 to 8 alphanumeric characters beginning with an alphabetic character other than a $ character.

ADDRESS = cua|030
is the channel unit address for the BSC line represented by this LINE statement. The defined address can be overridden by the U= operand of the VARY ACT command.

ISTATUS = ACTIVE|INACTIVE
determines whether this line is activated when the channel – attachment major node to which it belongs is activated and provides a sifting value for the CLUSTER and TERMINAL statements following this line. You can override this with the SCOPE operand on the VARY ACT command when the major node is activated.

RETRIES = n|7

determines how many times VTAM tries to recover from errors that occur during transmission to or from the physical unit represented by this LINE statement.

n is an integer from 0 through 255.

SERVLIM = n|4

is the maximum number of output operations (WRITEs) that will be allowed before a polling operation is started.

n

n is an integer from 0 through 255. The value selected for *n* is application dependent and must consider the amount of data sent with each output operation. As a guide, the value should allow a polling operation to occur, in order to solicit operator input, before subsequent output operations overwrite the 3270 screen.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

VSE VM The CLUSTER Statement for a BSC Cluster Controller

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

The CLUSTER statement defines:

- The resource name of the cluster controller
- The type of station
- The general polling character of the station
- Certain procedural options VTAM is to use when communicating with the station.

Name	Definition Statement	Operands
name	CLUSTER	GPOLL=char [,CUTYPE=**3271**|3275] [,DLOGMOD=default logmode entry name][1] [,FEATUR2=([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])][1] [,ISTATUS=**ACTIVE**|INACTIVE][2] [,LOGAPPL=application program name][1] [,LOGTAB=interpret table name][1] [,MODETAB=logon mode table name][1] [,SPAN=spanname] [,TERM=3275|3277|3284|3286][1] [,USSTAB=USS definition table][1]

Notes:

1. *Applies to a TERMINAL definition statement.*
2. *You can code this operand in either the GROUP, LINE, CLUSTER, or TERMINAL statement. It is meaningful for lines, clusters, and terminals. Its explanation appears in the LINE, CLUSTER, and TERMINAL statement descriptions.*

name
is a minor node name for the cluster controller.

GPOLL=char
is the general polling character assigned to the cluster controller.

char
is the 2–digit hexadecimal representation of a single EBCDIC polling character.

Refer to Figure 25 on page 134 to find the right value for GPOLL.

Example: If the controller address is A, code GPOLL=C1.

CUTYPE = 3271|3275
tells whether the controller of the station is a 3271 or 3275 BSC controller.

Note: Code **3271** for a BSC 3274, 3276, or 5937 controller. Code **3275** for a BSC 5275 controller.

ISTATUS = ACTIVE|INACTIVE
provides a "sifting" value for the TERMINAL statements following this CLUSTER. If ISTATUS = ACTIVE, a cluster is automatically activated when the line to which it is attached is activated. Note that you can override this with the SCOPE operand on the VARY ACT command when the major node or the line is activated.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

If CU or Device Number is	CU or Device Address is	Code GPOLL= or ADDR=
0	SP [1]	40
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
29	)	5D
30	;	5E
31	¬	5F

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

Figure 25. BSC 3270 General Polling (GPOLL) and Device Selection (ADDR) Characters

VSE VM The TERMINAL Statement for a BSC Terminal

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

The TERMINAL statement defines:

- The resource name of the terminal
- The type of terminal
- The terminal features
- The device address (device selection character) by which VTAM will contact the terminal (multipoint line control)
- Certain procedural options VTAM is to use when communicating with the terminal.

A printer attached to a 3275 is defined by coding the FEATUR2=PRINTR operand in the TERMINAL statement definition statement for the 3275, and not by coding a separate TERMINAL statement for the printer.

Name	Definition Statement	Operands
name	TERMINAL	ADDR=char ,TERM=3275|3277|3284|3286 [,DLOGMOD=default logmode entry name] [,FEATUR2=([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])] [,ISTATUS=**ACTIVE**|INACTIVE][1] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,SPAN=spanname] [,USSTAB=USS definition table name]

Notes:

1. *You can code this operand in either the GROUP, LINE, CLUSTER, or TERMINAL statement. It is meaningful for lines, clusters, and terminals. Its explanation appears in the LINE, CLUSTER, and TERMINAL statement descriptions.*

name
is the minor node name of the logical unit represented by this statement. This operand is required.

ADDR=char
is the device address (device selection character) assigned to this terminal.

char
is the 2–digit hexadecimal representation of a single EBCDIC component selection. Refer to Figure 25 on page 134 to find the right value for ADDR.

Example: If the terminal's device selection character is 'C', code ADDR=C3.

TERM = 3275|3277|3284|3286
identifies the specific type of device represented by this TERMINAL definition statement. Code 3277 for a BSC 3278; code 3286 for a 3287 or 3288.

DLOGMOD = default logmode entry name
is the name of the logmode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logmode entry is not otherwise provided, the first entry in the applicable logmode table is used by default.

FEATUR2 = ([operand] [,operand] . . .)
identifies the machine features for this terminal.

<u>MODEL1</u>|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.

PRINTR|NOPRINTR
tells VTAM whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only for a 3275 terminal.

ISTATUS = <u>ACTIVE</u>|INACTIVE
determines whether this terminal is to be activated when the cluster controller is first activated. Note that this can be overridden by the SCOPE operand on the VARY ACT command when the major node is activated.

LOGAPPL = application program name
identifies an application program to which this terminal is to be automatically logged on when the terminal is activated. The name must correspond to the network unique name assigned to the application program by an APPL definition statement.

If you do not code LOGAPPL, logons are initiated through USS facilities by the terminal operator, an application program, or the domain operator.

LOGTAB = interpret table name
is the name of an interpret table to be used by VTAM when processing logons originating from the terminal. See *VTAM Customization.* for a description of how an interpret table is defined.

MODETAB = logon mode table
is the name of a logon mode table to be used for the terminal. Logon mode tables are described in *VTAM Customization.* If this operand is omitted, the IBM – supplied logon mode table is used for the terminal.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

USSTAB = USS definition table name
is the name of a USS definition table to be used for the terminal. USS definition tables are described in *VTAM Customization.* If this operand is omitted, the IBM – supplied USS definition table is used for the terminal when character – coded requests are received.

Defining Local Non – SNA Major Nodes

A local non – SNA major node is defined by an LBUILD statement followed by a collection of VTAM definition statements defining each channel – attached non – SNA terminal as part of a logical set (group) of channel – attached non – SNA terminals. The controller to which the terminal is attached is not defined to VTAM. Terminals defined within a local non – SNA major node need not all be on the same controller. Different terminals on the same controller can be defined to VTAM in different local non – SNA major nodes. See Figure 26 for an example of statements defining local non – SNA major nodes.

You do not need to code a definition statement for the non – SNA cluster controller (3272 or compatible device). Non – SNA terminals connected to a single cluster controller do not have to be defined as a single major node.

```
LC3270NS LBUILD
LOC3277  LOCAL  CUADDR=00A,TERM=3277,DLOGMOD=S3270,
                LOGAPPL=AACCTS
```

Figure 26. An Example of Statements Defining a Local non – SNA Major Node

The LBUILD Statement

You must code one LBUILD statement each logical group (major node) of channel-attached non-SNA terminals. Define each terminal (minor node) in the group by using a LOCAL definition statement.

The format of the LBUILD statement is:

Name	Definition Statement	Operands
[name]	LBUILD	[CONFGDS=name] [,CONFGPW=password]

name
name is optional.

CONFGDS=name
is a 1- to 8-character data definition name that identifies the configuration restart data set defined by the user to be used for this major node. Include a DD statement with this data definition name in the VTAM start procedure.

Refer to *Network Program Products Planning* for a discussion of configuration restart.

CONFGPW=password
is the 1- to 7-character password, if required, for VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM prompts the VTAM operator for the correct password when VTAM attempts to use the data set.

You may code CONFGPW only if you also code CONFGDS.

VM Both CONFGDS and CONFGPW are ignored if coded on a VM system.

Local Non-SNA Major Node: LOCAL

The LOCAL Definition Statement

One or more LOCAL definition statements can be grouped with an LBUILD statement.

Code one LOCAL statement for each channel-attached non-SNA terminal that is in the VTAM domain.

The format of the LOCAL definition statement is:

Name	Definition Statement	Operands
name	LOCAL	CUADDR=channel device address ,TERM=3277|3284|3286 [,DLOGMOD=default logmode entry name] [,FEATUR2=([EDATS|**NOEDATS**] [,**MODEL1**|MODEL2])] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,SPAN=(spanname)] [,USSTAB=USS definition table name]

name
is the unique minor node name assigned to the device whose address appears in the CUADDR operand in this LOCAL definition statement.

CUADDR=channel device address
consists of three hexadecimal digits that identify the channel unit address for this channel-attached terminal. If CUADDR is omitted, you must code ISTATUS=INACTIVE and include the address in the VARY command that activates the unit. Note that you must *not* enclose the address within quotation marks or apostrophes.

TERM=3277|3284|3286
is the specific, channel-attached non-SNA terminal (printer or display station component). You can only code 3277, 3284, or 3286.

DLOGMOD=default logon mode table entry name
is the name of the logon mode entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logon mode entry is not otherwise provided, the first entry in the applicable logon mode table (given on the MODETAB operand or defaulted) is used by default. For more information on logon mode entries, see *VTAM Customization.*

FEATUR2=([EDATS|NOEDATS][,MODEL1|MODEL2])
identifies the machine features for a specific, channel-attached non-SNA terminal. These values are optional. The operands are listed below.

EDATS|NOEDATS
tells VTAM whether this terminal has the extended data stream feature.

MODEL1|MODEL2
is the specific model number (Model 1 or 2) for this 3277, 3284, or 3286 component. Code MODEL1 for those devices that have a default screen or buffer size of 480 bytes. Code Model 2 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 channel-attached terminal. These characteristics are obtained using the INQUIRE macro instruction. For more information, see *VTAM Programming.*

Note: The preferred way to communicate information about the defaults to be used for a session is with session parameters. See *Network Program Products Planning* for a discussion on session parameter sources.

ISTATUS=ACTIVE|INACTIVE
determines whether this terminal (minor node) is to be initially active when the local non-SNA major node to which it belongs is first activated. (Major nodes can be activated either when VTAM is started or, following the start of VTAM, by issuing the VARY command.)

When ISTATUS=ACTIVE is coded or assumed by default, this terminal is activated as part of the major node activation. If specified, automatic logon for this terminal to an application program occurs when the local non-SNA major node and application program are both active.

Once ISTATUS=INACTIVE is coded, the terminal is left inactive after the first activation of an inactive major node that contains this terminal. Later activations of an already active major node activate all the terminals not previously active (including the ones for which you have coded ISTATUS=INACTIVE).

Note: You can override the ISTATUS operand when the major node containing this definition is activated. This is done using the SCOPE operand on the VARY ACT command. For more information, see VTAM Operation.

LOGAPPL = application program name
identifies an application program (1- to 8- alphanumeric characters) to which a logon is to be automatically generated, on behalf of this terminal, when it is activated.

In VTAM V3R1.1, you can cancel this automatic logon definition at any time with the VARY NOLOGON command.

LOGTAB = interpret table name
is the member name of an interpret table.

An interpret table can be used to associate a logon request with the name of an application program or routine to which this logon is to be sent. See *VTAM Customization* for coding requirements.

MODETAB = logon mode table name
is the name of the logon mode table used to correlate each logon mode name with a set of session parameters. This name given for the MODETAB operand must be the name of a logon mode table created as described in *VTAM Customization.*

If this operand is not coded, an IBM-supplied logon mode table is used. The user can replace or modify the IBM-supplied logon mode table and rename it. This user-written table must be assembled, link-edited, and placed in VTAMLIB.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

USSTAB = USS definition table name
is the name of a USS definition table created as described in *VTAM Customization.* If you do not code USSTAB, an IBM-supplied USS definition table (ISTINCDT) is searched when character-coded input is received by VTAM from a channel-attached non-SNA terminal.

Defining Local SNA Major Nodes

A local SNA major node is defined by filing a single VBUILD statement for the major node and a separate PU or LU statement for each minor node. One VBUILD statement, placed before the first PU statement, must be included for each major node.

A PU statement is required for each physical unit (such as a cluster controller) in the major node. An LU statement for each logical unit is placed after the associated PU statement. Note that a physical unit and all its logical units must be defined within a single major node.

The PU and LU statements used to define a local SNA major node are very similar to the PU and LU statements that define a switched major node and to the PU and LU statements that define an NCP major node. The statements used for the local SNA major node are called PU (local) and LU (local) statements. See Figure 27 for an example of statements defining a local SNA major node.

```
LOCALSNA   VBUILD TYPE=LOCAL
LPU3790    PU     CUADDR=080,MODETAB=TAB3791,SSCPFM=FSS,
                  VPACING=2
LOC3790A   LU     LOCADDR=1,LOGAPPL=APAYR0
LOC3790B   LU     LOCADDR=2,LOGAPPL=AACCTS
LOC3790C   LU     LOCADDR=3,LOGAPPL=AACCTS
```

Figure 27. An Example of Statements Defining a Local SNA Major Node

VSE VM Note that PU and LU definition statements can be used to define devices that are attached to the host processor through a loop adapter. SNA devices, such as the 8775 and the 3276, are defined as if they were channel–attached physical units; there is a PU statement representing the controller and an LU statement for each associated logical unit. Non–SNA devices, such as the 3641 and the 3647, are defined as logical units associated with a single PU statement that represents physical unit functions provided by the loop adapter.

In the 4331 processor, the physical unit functions for non–SNA devices are assigned a channel unit address of 040. The 3644 is a special case: Although it is a non–SNA device, each 3644 must be defined with a PU definition statement. The 3644 and SNA physical units are assigned a channel unit address from 041 through 07F. The 3644 PU definition statement also may be followed by more than one LU definition statement. The definition statements starting with LPU3790 in Figure 27 represent a 3790 and its logical units that are attached to a loop adapter. For more information, see the appropriate device publications.

Local SNA Major Node: VBUILD

The VBUILD Statement

Code a VBUILD statement for each set of channel–attached SNA devices.

Code the VBUILD statement like this:

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=LOCAL [,CONFGDS=name] [,CONFGPW=password]

name
name is optional.

TYPE=LOCAL
means that the VBUILD statement defines a local SNA major node to VTAM. This operand is required.

CONFGDS=name
is a 1– to 8–character data definition name that identifies the configuration restart data set defined by the user to be used for this major node. Include a DD statement with this operand in the VTAM start procedure. Refer to *Network Program Products Planning* for a description of configuration restart.

CONFGPW=password
is the 1– to 7–character alphanumeric password used by VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM will prompt the VTAM operator for the correct password when VTAM attempts to open the data set.

You may code CONFGPW only if you also code CONFGDS.

VM Both CONFGDS and CONFGPW are ignored if coded on a VM system.

The PU (Local) Definition Statement

Code a PU definition statement for each physical unit in the local SNA major node.

Code the PU statement like this:

Name	Definition Statement	Operands
name	PU	[CUADDR=channel device address] [,DISCNT=([YES|NO][,F|NF])] [,DLOGMOD=default logmode entry name][1] [,ENCR=REQD|SEL|OPT|NONE][1] MVS [,ISTATUS=ACTIVE|INACTIVE][2] [,LOGAPPL=application program name][1] [,LOGTAB=interpret table name][1] [,MAXBFRU=n|1] [,MODETAB=logon mode table name][1] [,PACING=n|0|1][1] [,PUTYPE=2] [,SECNET=YES|NO] [,SPAN=(spanname)] [,SSCPFM=FSS|USSSCS][1] [,USSTAB=USS definition table name][1] [,VPACING=n|0|1][1]

Notes:

1. *You can code these operands in either the PU or the LU statement. They are meaningful only for LUs, but you can code them on the PU statement to take advantage of the sift–down effect. (See "Sift–Down Effect in VTAM Definition Statements" on page 71 for more information.) Explanations for these operands appear in the LU statement description on page 148.*

2. *You can code this operand in either the PU or the LU statement. It is meaningful for both physical units and logical units. Its explanation appears in the PU and LU statement description.*

name
provides the minor node name of the physical unit.

CUADDR=channel device address
is the hexadecimal channel device address to be used when activating the physical unit. If CUADDR is omitted, you must code ISTATUS=INACTIVE and include the address in the VARY command that activates the 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.

DISCNT=([YES|NO][,F|NF])

YES|NO

tells VTAM whether to terminate its SSCP–LU and SSCP–PU sessions when the last LU–LU session (involving the logical unit associated with this physical unit) is terminated. The device remains assigned to VTAM and is still active. (That is, sessions can be requested with its LUs; such a request causes SSCP–PU and SSCP–LU sessions to be reestablished. The physical unit itself may also request an SSCP–PU session.)

If you code YES, VTAM is to automatically terminate the SSCP–LU and SSCP–PU sessions as soon as the last LU–LU session is terminated. If any of the logical units, or the physical unit, request their own session termination, VTAM ignores the HOLD part of a character–coded logoff or the LAST–NOTLAST part of a field–formatted Terminate Self request. VTAM also rejects any attempt made by the physical unit to terminate its own SSCP session with a Request Discontact request.

If you code NO, VTAM is to terminate the SSCP–LU and SSCP–PU sessions when one of the following conditions is met:

VTAM receives a normal Request Discontact request from the physical unit, and all the LU–LU sessions associated with that physical unit have been terminated. All other existing LU–LU sessions are allowed to end normally.

VTAM receives an immediate Request Discontact request from the physical unit and immediately terminates any existing LU–LU sessions associated with that physical unit. This command overrides any previous normal Request Discontact request from the physical unit.

All LU–LU sessions have been terminated as a result of either a character–coded logoff from the SLU with HOLD=NO, a Terminate Self request with LAST specified, or a VARY INACT or TERM command from the operator, or a Cross–Domain Takedown request, or a TERMSESS macro. If any LU–LU sessions were terminated by any other means (for example, by an application program CLSDST macro instruction), the SSCP–LU and SSCP–PU sessions are not terminated.

F|NF
tells VTAM whether to indicate "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 take the default, "final – use" status is indicated. If you code NF, "not – final – use" status is indicated. 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.

ISTATUS = ACTIVE|INACTIVE
tells VTAM whether to activate the physical unit when its major node is activated following the first start of VTAM or a cold restart of VTAM (that is, a restart to initial status). It also provides a sifting value for the LU statements that follow this PU statement.

Note: The operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node. For more information, see *VTAM Operation.*

MAXBFRU = n|1
is the number of buffer units (elements of the IOBUF buffer pool) that will be used to receive data from the physical unit. *n* is a decimal integer.

PUTYPE = 2
is the physical unit type. Only PUTYPE = 2 (the default) is valid. Physical unit types are described in *SNA Reference Summary.*

SECNET = YES|NO
means that 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 management application.

Code this operand only if you have 3710s in your network.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Local SNA Major Node: LU

The LU (Local) Definition Statement

Code an LU definition statement for each logical unit associated with a physical unit within a local SNA major node. The LU statement must follow the PU statement that defines the physical unit with which this logical unit is associated.

Code the LU statement like this:

Name	Definition Statement	Operands
name	LU	LOCADDR=n [,DLOGMOD=default logmode entry name][1] [,ENCR=REQD|SEL|OPT|**NONE**][1] MVS [,ISTATUS=**ACTIVE**|INACTIVE][2] [,LOGAPPL=application program name][1] [,LOGTAB=interpret table name][1] [,MODETAB=logon mode table name][1] [,PACING=n|0|**1**][1] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS][1] [,USSTAB=USS definition table name][1] [,VPACING=n|0|**1**][1]

Notes:

1. *You can code these operands in either the PU or the LU statement. They are meaningful only for LUs, but you can code them on the PU statement to take advantage of the sift-down effect. (See "Sift-Down Effect in VTAM Definition Statements" on page 71 for more information.) Explanations for these operands appear in the LU statement description.*

2. *You can code this operand in either the PU or the LU statement. It is meaningful for both physical units and logical units. Its explanation appears in the PU and LU statement description.*

name
provides the minor node name of the logical unit.

LOCADDR=n
is the logical unit's local address at the physical unit. *n* is a decimal integer between 1 and 255.

An LU 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 the physical unit are not assigned network resources.

DLOGMOD=default logon mode table entry name
is the name of the logon mode table entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logon mode table entry is not otherwise provided, the first entry in the applicable logon mode table (given on the MODETAB operand or defaulted) is used by default.

ENCR=REQD|SEL|OPT|NONE MVS
tells whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD
means 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.

SEL
has no meaning for LUs. If ENCR=SEL is entered, ENCR=OPT is used instead.

OPT
means that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

NONE
means that this logical unit cannot engage in cryptographic sessions.

ISTATUS=ACTIVE|INACTIVE
tells VTAM whether to activate the logical unit automatically when the physical unit is activated following the first activation of the major node.

If ISTATUS=INACTIVE, this logical unit is left inactive after the first activation of its inactive major node. Later activations of an already active major node activate all logical units not previously active (including the ones for which you have coded ISTATUS=INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

LOGAPPL=application program name
is the name of an application program to which a logon request is to be automatically generated on behalf of the logical unit when the logical unit is activated. *application program name* can also be a name you have assigned with a USERVAR definition.

In VTAM V3R1.1, you can cancel this automatic logon definition at any time with the VARY NOLOGON command.

LOGTAB=interpret table name
is the name of an interpret table to be used by VTAM when processing logon requests that come from the logical unit. The interpret table named by LOGTAB is used to interpret the APPLID portion of an Initiate Self or character-coded logon request, as described in *VTAM Customization.*

MODETAB=logon mode table name
is the 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 for the MODETAB operand must be the name of a logon mode table created as described in *VTAM Customization*.

If you do not name a logon mode table for a logical unit by the MODETAB operand on either the PU or the LU statement, an IBM-supplied logon mode table (ISTINCLM) is used. The user can replace or modify the IBM-supplied logon mode table.

PACING=n|0|1
determines how VTAM is to pace the flow of data from the boundary node, which performs pacing for the channel-attached SNA device, to the secondary logical unit (SLU). If the primary logical unit (PLU), with which this SLU is in session, is in another domain, the boundary node is still the host in this domain. In this case, 2-stage pacing can be done between the SLU and VTAM, and between VTAM and the host containing the PLU. If the SLU, and the PLU with which this SLU is in session, are in the same domain, VTAM is the boundary node and the SRCVPAC operand in the MODEENT macro instruction (used to describe the session parameters in the logon mode table) is used to pace the flow of data directly from VTAM to the SLU (1-stage pacing). If SRCVPAC is zero, the VPACING value is used. When the SLU and the PLU are in the same domain, the PACING value is ignored. For more information on pacing, see "Defining Session Pacing Values" on page 76.

n
is the number of normal-flow requests that VTAM is to send 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. This value is ignored if the PLU and the SLU are in the same domain.

n is a decimal integer in the range 1 through 63.

0
means that no pacing is to be performed for sessions with the logical unit.

If you omit the PACING operand, the default is PACING=1.

SPAN=(spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide*.

SSCPFM = FSS|USSSCS

determines whether a logical unit can support character-coded messages (SSCPFM = USSSCS) in its communications with the SSCP. If not, only formatted commands (SSCPFM = FSS) are used.

See the publications for each individual device to see if formatted or character-coded messages are supported by that device.

USSTAB = USS definition table name

is the name of a USS definition table that is created as described in *VTAM Customization*. If you do not code USSTAB, an IBM-supplied USS definition table (ISTINCDT) is searched when character-coded input is received by VTAM from a logical unit.

VPACING = n|0|1

determines how VTAM is to pace the flow of data from the boundary node containing the primary logical unit (PLU) to the boundary node that performs pacing for the channel-attached SNA device. If the PLU is in another domain, the host in this domain is the boundary node for the secondary logical unit (SLU) and the host in the other domain contains the boundary node for the PLU. If the SLU and the PLU are in the same domain, VTAM ignores the PACING value and uses the VPACING value to pace the data flow directly from VTAM to the SLU (1-stage pacing). For more information on pacing, see "Defining Session Pacing Values" on page 76.

n

is the number of normal-flow requests that the PLU is to send for a given LU-LU session before waiting for a pacing response. No further normal-flow requests can be sent to the LU until it is ready to receive them.

n is a decimal integer from 1 through 63.

0

means that no pacing is to be performed for sessions with the logical unit.

If you omit the VPACING operand, the default is VPACING = 1.

Defining Switched Major Nodes

A switched major node is defined by filing a single VBUILD statement for the major node and separate PU, PATH, and LU statements for each minor node. One VBUILD statement must be included in each member, placed before the first PU statement.

The PU and LU statements define physical units and logical units, on switched SDLC lines, that can be dialed into or dialed out from communication controllers. (These SDLC lines are defined in switched line groups in the NCP major node.) For dial-out operations, the PATH statement defines the possible paths to be used to establish connection between the communication controller and the physical unit. Additionally, PU and LU statements that define virtual physical units and logical units with dial-in capability (for example, NTO-controlled devices), should be coded in a switched major node. Virtual physical units and logical units can only be coded for switched lines defined in an NCP major node.

The PU and LU statements used in a switched major node are very similar to the PU and LU statements used in a local SNA major node, and to the PU and LU statements used in an NCP major node. The statements used for the switched major nodes are called PU (switched) and LU (switched) statements. See Figure 28 on page 153 for an example of statements defining a switched major node. For planning information on switched major nodes, see *Network Program Products Planning.*

For information on defining a controlling PLU for a switched SNA device in an interconnected network, see "Defining a Cross-Network Controlling Primary Logical Unit" on page 249.

```
SWNODE1    VBUILD TYPE=SWNET,MAXGRP=1,MAXNO=2
SWPU3790   PU     ADDR=05,                 SDLC ADDRESS OF STATION
                  IDBLK=006,
                  IDNUM=19781,
                  PUTYPE=2,                SNA CLUSTER CONTROLLER (PU_T2)
                  MAXPATH=2,
                  MAXDATA=265,             BIGGEST FRAME 3790 CAN ACCEPT
                  MAXOUT=7,                VTAM CAN SEND 7 FRAMES TO 3790
                  SSCPFM=FSS               USS NOT INVOLVED IN LOGONS
SW01       PATH   DIALNO=19288324,         TELEPHONE NUMBER OF 3790
                  GID=1,                   SWITCHED LINE GROUP IDENTIFIER
                  GRPNM=HGROUP40,          SWITCHED LINE GROUP NAME
                  PID=1                    DIAL PATH IDENTIFIER
SW02       PATH   DIALNO=12183392456,      TELEPHONE NUMBER OF 3790
                  GID=1,                   SWITCHED LINE GROUP IDENTIFIER
                  GRPNM=HGROUP40,          SWITCHED LINE GROUP NAME
                  PID=2                    DIAL PATH IDENTIFIER
IN3790LU   LU     LOCADDR=1,               PU_T2 ORIGIN IS 1. LU 'IN' FOR SSS
                  PACING=1,
                  LOGAPPL=SYSSSS,
                  ISTATUS=INACTIVE         WILL NOT BE ACTIVATED AUTOMATICALLY
LU379002   LU     LOCADDR=2,
                  LOGAPPL=AINV0,
                  PACING=1
LU379003   LU     LOCADDR=3,
                  LOGAPPL=AINV0,
                  PACING=1
```

Figure 28. An Example of Statements Defining a Switched Major Node

Switched Major Node: VBUILD

The VBUILD Statement

Code a VBUILD statement for each switched major node.

Code the VBUILD statement like this:

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=SWNET [,CONFGDS=name] [,CONFGPW=password] [,MAXGRP=n] [,MAXNO=n]

name
The *name* is optional.

TYPE=SWNET
means that the VBUILD statement defines a switched major node to VTAM. All physical units defined in this major node can be connected only by means of a switched link. This operand is required.

CONFGDS=name
is a 1- to 8-character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the VTAM start procedure.

Refer to *Network Program Products Planning* for a description of configuration restart.

VM This operand is ignored if coded on a VM system.

CONFGPW=password
is the 1- to 7-character password, if required, for VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM prompts the operator for the correct password when VTAM attempts to open the data set.

You may code CONFGPW only if you also code CONFGDS.

VM This operand is ignored if coded on a VM system.

MAXGRP=n
is the number of unique path groups (GROUP names) that are defined in the GRPNM operand of all PATH statements within the switched major node. The maximum value of *n* is 32767. (Code this operand only if you also code the switched PATH statement.)

MAXNO=n
is the number of unique telephone numbers that are defined in the DIALNO operand of all PATH statements within the switched major node. The maximum value of *n* is 32767. (Code this operand only if you also code the switched PATH statement.)

The PU (Switched) Definition Statement

Code a PU definition statement for each physical unit in the switched major node.

Code the PU statement like this:

Name	Definition Statement	Operands
name	PU	ADDR = station address ,IDBLK = identification block ,IDNUM = identification number [,BATCH = YES|**NO**][1] [,DISCNT = (YES|**NO**][,**F**|NF])] [,DLOGMOD = default logmode entry name][1] [,ENCR = REQD|SEL|OPT|**NONE**][1] MVS [,FEATUR2 = LOWERCSE|**DUALCSE**][1] [,IRETRY = YES|**NO**] [,ISTATUS = **ACTIVE**|INACTIVE][2] [,LOGAPPL = application program name][1] [,LOGTAB = interpret table name][1] [,MAXDATA = size|**261**|**265**] [,MAXOUT = n|**1**] [,MAXPATH = n|**0**] [,MODETAB = logon mode table name][1] [,PACING = n|0|**1**][1] [,PASSLIM = n|**1**] [,PUTYPE = n|**2**] [,SECNET = YES|**NO**] [,SPAN = (spanname)] [,SSCPFM = **FSS**|USSSCS|USSNTO| USS3780|USS3270|USS3275][1] [,TERM = terminal type][1] [,USSTAB = USS definition table name][1] [,VPACING = n|0|**2**][1]

Notes:

1. *You can code these operands in either the PU or the LU statement. They are meaningful only for LUs, but you can code them on the PU statement to take advantage of the sift-down effect. (See "Sift-Down Effect in VTAM Definition Statements" on page 71 for more information.) Explanations for these operands appear in the LU statement description on page 160.*

2. *You can code this operand in either the PU or the LU statement. It is meaningful for both physical units and logical units. Its explanation appears in the PU and LU statement description.*

name
provides the minor node name of the physical unit.

ADDR = station address
is the 8-bit SDLC station address for the physical unit and is required. You can get this address from the person who planned the device's installation. Code the address in hexadecimal and do not enclose it in quotation marks or apostrophes.

IDBLK = identification block

is a three – digit hexadecimal number that denotes the device type. Do not enclose it in quotation marks or apostrophes. You can get the number from the component description manual for the device or from the person who planned the device's installation. (Component description manuals are listed in *Network Program Products Bibliography and Master Index.*)

The block number occupies bits 16 through 27 of the 48 – bit station ID constructed by VTAM for switched network operation. (See the IDNUM operand below for a description of the station ID.)

IDBLK is required; you must code it.

IDNUM = identification number

is a five – digit hexadecimal number that identifies the specific device. The device's serial number is often used for this purpose. You can get this number from the component description manual for the device or from the person who planned the device's installation. (Component description manuals are listed in *Network Program Products Bibliography and Master Index.*)

IDNUM is required; you must code it. See *Network Program Products Planning* for special NTO considerations when defining a virtual physical unit.

VTAM uses IDBLK and IDNUM to build a 48 – bit station ID that is used in XID exchange during the dial procedure. This station ID must be unique for each station within the network (not just within the major node).

The station ID is structured as follows:

Bits	Meaning
0 – 3	Reserved
4 – 7	PUTYPE
8 – 15	"00"
16 – 27	IDBLK
28 – 47	IDNUM

DISCNT = ([YES|<u>NO</u>][,<u>F</u>|NF])

YES|<u>NO</u>

tells VTAM whether to physically disconnect the physical unit when the last logical unit terminates 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, as contrasted with a leased link, does not involve the deactivating the physical unit or its logical units. (That is, sessions can be requested with those logical units; such a request causes the physical connection to be reestablished.)

If you code YES, VTAM is to automatically disconnect the PU as soon as the last logical unit terminates its session with its application program. If any logical units request their own session termination, VTAM ignores the part of their request that determines whether the physical unit is to be disconnected (that is, the HOLD part of a character−coded logoff request or the LAST−NOLAST part of a field−formatted Terminate Self request is ignored). VTAM also rejects any attempt made by the physical unit to request its own disconnection (using the Request Discontact request).

If you code NO, VTAM is to disconnect the physical unit when one of the following conditions is met:

VTAM receives a normal Request Discontact request from the PU and all LU−LU sessions have been terminated. Other existing sessions are allowed to end normally.

VTAM receives an immediate Request Discontact request from the physical unit. VTAM immediately terminates existing LU−LU sessions. This request overrides any previous normal Request Discontact request from the physical unit.

All LU−LU sessions have been terminated as a result of a character−coded logoff request with HOLD=NO, a Terminate Self request for which LAST was specified, or a VARY INACT command. If any LU−LU session was terminated by any other means (such as by an application program CLSDST), the PU is not disconnected.

F|NF
tells VTAM whether to indicate “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 take the default, “final−use” status is indicated. If you code NF, “not−final−use” status is indicated. 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.

IRETRY=YES|NO
determines whether the boundary NCP (the NCP to which the switched PU will become connected) is to retry a polling operation immediately for the device if an Idle Detect Timeout follows a polling operation. For more information on this operand, see *NCP−SSP Resource Definition Reference.*

ISTATUS = ACTIVE|INACTIVE
determines whether the physical unit is to be activated after the first activation of the switched major node. ISTATUS also provides a sifting value for the LU statements that follow this PU statement. If ISTATUS = INACTIVE, this physical unit is left inactive after the first activation of an inactive major node that contains the LINE for this physical unit. Later activations of an already active major node activate all the lines not previously active and their physical units (including the ones for which you have coded ISTATUS = INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

MAXDATA = size|261|265
is the maximum amount of data in bytes—including the transmission header (TH) and request/response header (RH)—that the physical unit can receive in one segment of a path information unit (PIU). *size* is a decimal integer. The maximum value is 65535 bytes. For physical units connected through an NCP, see *NCP–SSP Resource Definition Reference* for more information. If you do not code MAXDATA, the value of 261 bytes is used for a PU type 1 and 265 for a PU type 2.

MAXOUT = n|1
is the maximum number of path information units (PIUs) that will be sent to the physical unit before requesting a response from it. *n* is a decimal integer from 1 through 7.

MAXPATH = n|0
is the number of dial–out paths to the physical unit. *n* is a decimal integer from 0 through 256. Zero means that only dial–in paths are available.

Refer to the switched PATH definition statement for a description of defining a dial–out path to a physical unit.

PASSLIM = n|1
determines, for switched physical units connected through an NCP only, the maximum number of contiguous path information units (PIUs) that the NCP will send to the physical unit at once. *n* is a decimal integer between 1 and the value of MAXOUT.

PUTYPE = n|2
is the physical unit type. The physical unit type depends on the type of cluster controller (such as 3767 or 3791). To determine the physical unit type for a given device, see the component description manual for that device.

The value of PUTYPE occupies bits 4 through 7 of the 48–bit station ID constructed by VTAM for switched network operation. (See the IDNUM operand above for a description of the station ID.)

SECNET = YES|NO

means that 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 management application.

Code this operand only if you have 3710s in your network.

SPAN = (spanname)

Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Switched Major Node: LU

The LU (Switched) Definition Statement

Code an LU definition statement for each logical unit associated with a physical unit within a switched major node. The LU statement must follow the PU statement that defines the physical unit with which the logical unit is associated.

Code the LU statement like this:

Name	Definition Statement	Operands
name	LU	LOCADDR=n [,BATCH=YES|NO][1] [,DLOGMOD=default logon mode table entry] [,ENCR=REQD|SEL|OPT|NONE][1] MVS [,FEATUR2=LOWERCSE|DUALCSE][1] [,ISTATUS=ACTIVE|INACTIVE][2] [,LOGAPPL=application program name][1] [,LOGTAB=interpret table name][1] [,MODETAB=logon mode table name][1] [,PACING=n|0|1][1] [,SPAN=(spanname)] [,SSCPFM=FSS|USSSCS|USSNTO| USS3780|USS3270|USS3275][1] [,TERM=terminal type][1] [,USSTAB=USS definition table name][1] [,VPACING=n|0|2][1]

Notes:

1. *You can code these operands in either the PU or the LU statement. They are meaningful only for LUs, but you can code them on the PU statement to take advantage of the sift-down effect. (See "Sift-Down Effect in VTAM Definition Statements" on page 71 for more information.) Explanations for these operands appear in the LU statement description.*

2. *You can code this operand in either the PU or the LU statement. It is meaningful for both physical units and logical units. Its explanation appears in the PU and LU statement description.*

name
provides the minor node name of the logical unit.

LOCADDR=n
is the logical unit's local address at the physical unit. *n* is a decimal integer.

The range of valid local addresses depends on the PUTYPE of the physical unit with which the logical unit is associated. PUTYPE=1 allows local addresses from 0 to 63; PUTYPE=2 allows local addresses from 1 to 255. To determine the PUTYPE and the limits on local addresses for a particular physical unit, see the component description manual for the device.

An LU 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.

BATCH = YES|NO

is the processing priority that the NCP is to use for the logical unit. BATCH = NO means a high priority (suitable for interactive application programs); BATCH = YES means a low priority.

DLOGMOD = default logon mode table entry name

is the name of the logon mode table entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logon mode table entry is not otherwise provided, the first entry in the applicable table (given on the MODETAB operand or defaulted) is used by default.

ENCR = REQD|SEL|OPT|NONE MVS

tells whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD

means that VTAM must encipher all messages to and from this logical unit. If ENCR = REQD is coded, no sessions can be established with this LU unless the host with which it is associated is able to handle cryptographic sessions.

SEL

has no meaning for LUs. If ENCR = SEL is entered, ENCR = OPT is used instead.

OPT

means that this logical unit is capable of engaging in cryptographic sessions, but allows the application program to determine whether to use cryptography.

NONE

means that this logical unit cannot engage in cryptographic sessions.

FEATUR2 = LOWERCSE|DUALCSE

contains, for virtual LUs that are supported by NTO, certain device information.

LOWERCSE|DUALCSE

tells VTAM how to send alphabetic characters coded with the TEXT operand on a USSMSG macro instruction to a non-SNA terminal over the SSCP-LU session. Nonalphabetic characters, as well as all characters coded using the BUFFER operand on a USSMSG macro instruction, are not affected by this operand.

LOWERCSE

means that alphabetic characters will be sent to the terminal over the SSCP-LU session in lowercase.

DUALCSE
means that VTAM will send all characters as they are coded in the USSMSG macro instruction. If you do not code either value, DUALCSE is assumed.

ISTATUS = ACTIVE|INACTIVE
determines whether the logical unit is to be activated automatically when the physical unit is activated.

If ISTATUS = INACTIVE, this logical unit is left inactive after the first activation of its inactive major node. Later activations of an already active major node activate all logical units not previously active (including the ones for which you have coded ISTATUS = INACTIVE).

Note that the operator can override this value by using the SCOPE operand on the VARY ACT command when activating the major node.

LOGAPPL = application program name
is the name of an application program to which the logical unit is to be logged on automatically when the logical unit is activated.

For logical units that are accessible only by dial-in paths, the application program receives control after the dial-in connection is made. For dial-in/dial-out or dial-out paths, control is given to the application program immediately.

In VTAM V3R1.1, you can cancel this automatic logon definition at any time with the VARY NOLOGON command.

LOGTAB = interpret table name
is the name of an interpret table to be used by VTAM when processing logons that come from the logical unit. See *VTAM Customization* for a description of how an interpret table is defined.

MODETAB = logon mode table name
is the logon mode table to be used to correlate each logon mode name with a set of session parameters for the logical unit associated with this physical unit. The name must be the same name used for a logon mode table that is created as described in *VTAM Customization.*

If you do not name a logon mode table for a logical unit by the MODETAB operand in either the PU or the LU statement, an IBM-supplied logon mode table is used. The user can modify or replace the IBM-supplied logon mode table.

PACING = n|0|1
determines how pacing is to be handled between the logical unit and the boundary node to which it is connected. (In contrast, VPACING involves pacing between VTAM and the boundary node.) For more information on pacing, see "Defining Session Pacing Values" on page 76.

n

is the number of normal-flow RUs that the boundary node is to send to the logical unit before waiting for a pacing response.

n is a decimal integer from 1 through 63. For logical units connected through an NCP, refer to *NCP-SSP Resource Definition Reference* for information on how to select the proper value.

0

means that no pacing is to be performed for sessions with the logical unit.

If you omit the pacing operand, the default is PACING=1.

SPAN=(spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

SSCPFM=<u>FSS</u>|USSSCS|USSNTO|USS3780|USS3270|USS3275
determines whether the logical unit can support field-formatted (FSS) RUs, or character-coded (USSSCS) RUs over its SSCP-LU session. USSNTO means that this logical unit is supported by the NTO program product in the boundary node and that character-coded RUs over its SSCP-LU sessions are supported. FSS is the default.

TERM=terminal type
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 macro that names this virtual LU. The information is placed in the DEVAUXTP field of the DEVCHAR DSECT. (This 8-byte storage area is provided by the application program.) For more information on the DEVCHAR DSECT, see *VTAM Programming.*

USSTAB=USS definition table name
is the name of a USS definition table that is defined as described in *VTAM Customization.*

If you do not code USSTAB, the VTAM-supplied USS definition table (ISTINCDT) is searched when VTAM receives input from an logical unit.

VPACING = n|0|2

determines how VTAM is to pace the flow of data from VTAM to the boundary node to which the logical unit is connected. For more information on pacing, see "Defining Session Pacing Values" on page 76.

n

is the number of normal-flow RUs that VTAM is to send for a given LU-LU session before waiting for a pacing response. No more requests can be sent to the boundary node until it replies with a pacing response to VTAM to show that it is ready to receive more RUs.

n is a decimal integer from 1 through 63, and greater than the value of *n* used with the corresponding PACING operand.

0

means that no pacing is to be performed for sessions with the logical unit.

If the VPACING operand is omitted, VPACING = 2 is assumed in a switched major node.

The Switched PATH Definition Statement

The PATH definition statement is used to define a dial‑out path to a PU in a switched major node. Note that certain operands apply only to X.21 switched protocol. You can code as many PATH statements as you need, to a maximum of 256, for each physical unit. The PATH statement must immediately follow the PU statement that defines the associated physical unit. VTAM searches the PATH statements for an available path in the order given in the configuration deck.

Code the PATH statement like this:

Name	Definition Statement	Operands
name	PATH	{DIALNO = telephone number|LINENM = linename} [,GID = n] [,GRPNM = groupname] [,PID = n] [,REDIAL = n|**3**] [,SHOLD = YES|NO] VSE [,USE = **YES**|NO]

name

name is optional.

DIALNO = telephone number

is the telephone number (in EBCDIC) used to initiate a connection with a physical unit over a switched link. You can insert special characters in the number according to the following rules:

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

DIALNO is mutually exclusive with the LINENM operand for any single switched path statement. If SHOLD=YES, however, DIALNO is *required* and you can code it together with LINENM.

For more information on this operand, refer to *NCP‑SSP Resource Definition Reference*.

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 will supply 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 (hex 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.

Instead of coding:

```
P1 PATH DIALNO=5551235    FULL DIAL ADDRESS FOR PATH P1
```

you would code:

```
P1 PATH DIALNO=.27        ABBREVIATED DIAL ADDRESS FOR
                          PATH P1
```

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.

For more information on X.21 switched line features, see *General Information: X.21 Interface Features.*

For **NCP/Token-Ring interconnection**, use the following format when coding DIALNO for a terminal that is to be called by a host:

DIALNO = aabb4000cccccccc

where:

aa is the token-ring interface coupler (TIC) number of the communication controller (between 00 and 99).

bb is the system-access-point address of the terminal (a multiple of 4).

cccccccc is the last four bytes of the terminal's ring-station address (the first digit must be between 0 and 7).

aa, bb, and *cccccccc* all are decimal numbers.

LINENM = linename VSE

names a line for the direct call function. This line must be part of an X.21 line group and must be defined as a direct call line by the common carrier facility for the link–station whose dial–out path is being defined.

LINENM is mutually exclusive with the DIALNO operand for any single switched path statement. If SHOLD = YES, however, you can code DIALNO and LINENM together.

It is recommended, but not required, that the PATH statement for a direct call line be the first statement following the switched PU statement. This will ensure that VTAM will attempt to call the switched physical unit using the direct call line first.

GID = n

identifies a group of paths across all PUs in the switched major node. *n* is a decimal integer from 0 through 255.

Group identifiers can be assigned to let the VTAM operator 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 = groupname

is the symbolic name of a GROUP statement in an NCP 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 if SHOLD = YES. For more information on this operand, refer to *NCP–SSP Resource Definition Reference.*

PID = n

is an identifier for the path being defined. This identifier is unique for a given physical unit. The operator uses this identifier to change the status of the path. *n* is a decimal integer from 0 through 255.

REDIAL = n|3

is the number of times dialing is to be retried before returning a dialing error to VTAM. *n* is a decimal integer. The minimum value for *n* is 0, which means that dialing is not to be retried. The maximum value for *n* is 254.

SHOLD = YES|NO VSE

tells VTAM whether this path is an X.21 Short Hold Mode/Multiple Port Sharing (SHM/MPS) path. (X.21 Short Hold Mode/Multiple Port Sharing is a feature available with VTAM V3R1.)

SHOLD = YES means this path is an SHM/MPS path. VTAM uses the values of *free* and *npoll* coded for the SHOLD operand on the corresponding GROUP statement for the channel–attachment major node. (This operand is described on page 124.)

SHOLD = NO is the default and means this path is not an SHM/MPS path.

If you code SHOLD = YES, you must also code DIALNO and GRPNM on this PATH statement.

USE = YES|NO
tells VTAM whether to consider the path initially usable or not usable. This attribute of the path can be modified by the VTAM operator. 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.

Defining Network Control Program (NCP) Major Nodes

An NCP major node is defined by filing NCP definition statements 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. This requirement provides identification to the installation of which major node is affected when VTAM issues messages containing the NCP load module name.

Additional information, required by VTAM, to communicate with the NCP and its attached devices is provided to VTAM by coding VTAM – only definition statements and VTAM – only operands in NCP definition statements. The VTAM – only definition statements and operands must appear in the NCP generation definition statements that define this NCP to VTAM, even though they provide no information to the NCP.

If multiple copies of the NCP definition statements with different VTAM operand values are required by an installation at a single host, the sets can be filed under temporary names. Before you activate any one of the sets as a VTAM major node, the member must be renamed to match the load module ID on the BUILD definition statement.

Listed below are NCP definition statements used by VTAM in a correct coding sequence for NCP generation, along with an indication of whether they apply to VTAM, VTAM and NCP, or NCP with VTAM requirements.

PCCU	VTAM only
BUILD	Both VTAM and NCP
SYSCNTRL	Both VTAM and NCP
HOST	Both VTAM and NCP
LUDRPOOL	NCP only, with VTAM requirements
PUDRPOOL	NCP only, with VTAM requirements
GROUP	Both VTAM and NCP
LINE	Both VTAM and NCP
CLUSTER	Both VTAM and NCP
TERMINAL	Both VTAM and NCP
GROUP	Both VTAM and NCP
LINE	Both VTAM and NCP
PU	Both VTAM and NCP
LU	Both VTAM and NCP
NETWORK	Both VTAM and NCP
GWNAU	NCP only, with VTAM requirements
GENEND	Both, no VTAM requirements

The PCCU definition statement is for VTAM only and is described in "The PCCU Definition Statement" on page 171. The NETWORK and GWNAU statements are defined only for a gateway NCP.

If a definition statement is marked "Both VTAM and NCP," it means that VTAM processes this statement. 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 *NCP – SSP Resource Definition Reference.*

NCP Major Nodes

If a definition statement is marked "Both, no VTAM requirements," it means that both VTAM and the NCP use this statement, but it has no special VTAM requirements and should be coded as described in *NCP–SSP Resource Definition Reference.*

If a definition statement is marked "NCP only, with VTAM requirements," you will find its description in *NCP–SSP Resource Definition Reference*, but you should also refer to this book for special VTAM requirements.

Coding VTAM–Only Definition Statements and Operands

VTAM information contained in an NCP major node can occur in:

- The PCCU definition statement
- NCP definition statements for the NCP itself
- NCP definition statements for resources controlled by the NCP.

The PCCU Definition Statement

The PCCU (programmed communication control unit) definition statement identifies the communication controller into which the NCP is loaded. It is required in VTAM systems and defines the VTAM functions that are to be provided for this NCP. Code at least one PCCU statement for each NCP.

If more than one VTAM host will activate the NCP, you can code a PCCU statement for each host. Each PCCU statement describes the functions performed by the host it is associated with. For example, if you want to assign a second host to back up a communication management configuration (CMC) host, code a PCCU statement for the backup host. The statement is not used until a failure occurs, at which time the backup host activates the NCP. (For general information about backing up a CMC host, see "Defining Channel-Attached NCP Support" on page 101.)

The SUBAREA operand, described on page 178, associates 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 statement is necessary, but the operator may need to override unique attachment requirements.

Assigning Resources to More than One Host

Every resource attached to the NCP is owned by a VTAM host. The values you code for each NCP major node determine which host owns which resources. The value of the OWNER operand on the PCCU statement is matched with the OWNER operands on the GROUP and LINE statements in the NCP.

More than one host can share ownership of a line or a PU type 4 link station. You specify this in the NCP by coding the appropriate names as values of the OWNER operand on the GROUP or LINE statements. However, only one host can own a PU type 1 or PU type 2 and its associated logical units. In this situation, the first host to activate the physical unit becomes its owner. Then, other hosts cannot activate this physical unit.

Use the BACKUP operand on the PCCU statement to allow a backup host to take over NCP resources when there is a failure in the host that owns them. Code BACKUP=YES to identify all resources that this host can take over in case of a failure. When a failure occurs, an operator can issue a VTAM VARY ACQ command to acquire these resources without disrupting sessions.

The BACKUP and OWNER operands are described on pages 173 and 178, respectively. For information about the possible requirements for this function, see *Network Program Products Planning*.

NCP Major Node: PCCU

Format of the PCCU Statement

PCCU definition statements must be at the beginning of the NCP generation deck.

The format of the PCCU definition statement is:

Name	Definition Statement	Operands
name	PCCU	NCPLUB=lubname VSE [,AUTODMP=YES|**NO**] [,AUTOIPL=YES|**NO**] [,AUTOSYN=**YES**|NO] [,BACKUP=YES|**NO**] [,CDUMPDS=dumpname] [,CHANCON=**COND**|UNCOND][1] [,CONFGDS=name] [,CONFGPW=password] [,CUADDR=channel device address][1] [,DUMPDS=dumpname] [,DUMPSTA=link station name] [,GWCTL=ONLY|**SHR**][2] [,INITEST=YES|**NO**] [,LOADSTA=link station name] [,MAXDATA=size|**65535**][1] [,MDUMPDS=dumpname] [,NETID=host network id][2] [,OWNER=ownername] [,RNAME=link station name|(name, . . . ,name)] [,SUBAREA=n] [,VFYLM=YES|**NO**]

Notes:

1. *This operand is not used for data hosts defined in a channel-attachment major node. See descriptions in "Defining Channel-Attachment Major Nodes" on page 92 for replacement operands.*

2. *SNA network interconnection only.*

name
The *name* is optional.

NCPLUB=lubname VSE
is the symbolic name of the NCP load file that contains the NCP phase. The symbolic device name appears on the EXTENT statement for the NCP phase. See the sample procedure for starting VTAM in Figure 18 on page 66.

For more information, see "NCP Load File" on page 63.

AUTODMP=YES|NO
determines whether a dump of communication controller storage is taken automatically after a recoverable failure in the NCP or the communication controller. When AUTODMP=YES, the dump is taken automatically. When AUTODMP=NO, the VTAM operator is

asked whether a dump should be taken. After the dump, VTAM loads another copy of the same NCP and restarts it.

This operand is valid only if you code a DUMPDS operand in this PCCU definition statement. Otherwise, no NCP dump can be taken. (DUMPDS is described on page 175.)

In an NCP for a communication controller that is attached by multiple channels, only one PCCU statement should have AUTODMP=YES.

AUTOIPL=YES|NO

determines whether VTAM is to load another copy of the NCP and restart it after either:

- An unrecoverable failure occurs in the NCP or the communication controller, or
- A dump is taken of the controller.

In an NCP for a communication controller that is attached by multiple channels, only one PCCU statement should have AUTOIPL=YES.

AUTOSYN=YES|NO

causes the NCP, when activated, to tell VTAM the name of the NCP loaded in the communication controller. It applies only if the VTAM operator requests (or defaults) LOAD=U on the VARY ACT command for an NCP.

When the name matches VTAM's major node name for the NCP it is activating:

- If AUTOSYN=YES, VTAM automatically begins to work with the NCP.
- If AUTOSYN=NO, the operator is asked whether this NCP should be used or the communication controller should be refreshed with a new copy.

When the name does not match VTAM's major node name, the value of VFYLM determines whether the controller is reloaded automatically. (VFYLM is described on page 179.)

BACKUP=YES|NO

determines whether this NCP's resources can be taken over by a backup host in the event their owning host fails.

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 statement.) The host VTAM cannot use the resources until a VTAM VARY ACQ command is issued for this NCP.

BACKUP may be used for all NCPs. If you do not also code the OWNER operand, however, it is ignored. (OWNER is described on page 178.)

CDUMPDS = dumpname

is one of three operands (DUMPDS, CDUMPDS, and MDUMPDS) that name dump files for this communication controller. CDUMPDS applies only to a 3725 Communication Controller.

For more information, see the description of the DUMPDS operand on page 175.

CHANCON = COND|UNCOND

determines whether VTAM sends a conditional or an unconditional channel contact request to the NCP when the host wishes to activate the NCP over a channel.

If you do not code CHANCON, or if CHANCON = COND, the contact request is *conditional*: the NCP can reject it if the host's subarea number is already in use by another host (attached to another one of the NCPs channels).

If 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.

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 LINE statement for the channel-attachment major node. See "The LINE Definition Statement" on page 105 for more information.

CONFGDS = name

is the 1- to 8-character data definition name of the configuration restart data set to be used for this major node. Include a DD statement with this data definition name in the VTAM start procedure.

Refer to *Network Program Products Planning* for a description of configuration restart.

VM This operand is ignored if coded on a VM system.

CONFGPW = password

is the 1- to 7-character password that VTAM uses to gain access to the configuration restart data set. If you do not code CONFGPW, but VSAM requires it, VSAM asks the operator for the correct password when VTAM tries to open the data set.

VM This operand is ignored if coded on a VM system.

CUADDR = channel device address

is the three-digit channel unit address, in hexadecimal, 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, he should provide an address in the U operand when he issues the VTAM VARY ACT command.

If no channel device address or link station contact points are coded (with the CUADDR and RNAME operands of PCCU) 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.

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 statement for the channel–attachment major node. See "The LINE Definition Statement" on page 105 for more information.

DUMPDS = dumpname
is one of three operands (DUMPDS, CDUMPDS, and MDUMPDS) that name dump files for this communication controller. CDUMPDS and MDUMPDS apply only to an IBM 3725 Communication Controller. DUMPDS applies to any communication controller.

- DUMPDS names the file that contains a storage dump of a communication controller. For a 3725, 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 3725, CDUMPDS names the file that contains CSP dumps.
- For a 3725, MDUMPDS names the file that contains MOSS dumps.

MVS VSE *dumpname* must be on a DD or DLBL statement in the cataloged procedure for starting VTAM. This statement defines the data set or file that is to contain the data from a storage dump of a communication controller.

You must code DUMPDS if:

- AUTODMP = YES on this PCCU definition statement
- AUTODMP = NO on this PCCU definition statement and the VTAM operator will answer YES to the resulting prompt for a dump, or
- The VTAM operator will issue 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 as described in the *NCP–SSP Diagnosis Guide.*

Ordinarily the data in a dump file is overlaid by the next dump. To avoid this you can have a separate dump file for each communication controller in the VTAM system, or you can allocate one dump file for dumping channel–attached communication controllers and another for dumping link–attached communication controllers. Another way to preserve the data in an existing dump file is to use the DUMPDS operand on the MODIFY DUMP command to direct data to another file.

DUMPSTA = link station name

is 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.

Notes:

1. *VTAM chooses default link stations that are adjacent to the NCP and ready for the dump operation. Preference is given to channel link stations.*

2. *VTAM gives names to channel link stations by taking the channel device address of the communication controller and adding '−S'. For example, the channel device address 0C2 has the link station name of 0C2−S. You may designate the same link station for both dumping and loading.*

3. *If you have defined a channel−attachment major node whose major node and its link and link station minor nodes will be active when this NCP is activated, you must use the channel link station name (defined by the PU definition statement in the channel−attachment major node) for the DUMPSTA, LOADSTA, and RNAME operands of PCCU. You can also use the CUADDR operand instead of RNAME and use the automatically−defined link station name (such as 0C2−S; see item 2 above) for DUMPSTA and/or LOADSTA.*

   For the DUMPSTA, LOADSTA, and RNAME operands, you should use the channel link station name defined in the channel−attachment major node, provided (1) you have defined the channel unit address in the channel−attachment major node, and (2) that major node is active. This is not required, but it is recommended.

For more information on link stations and use of the VTAM VARY ACT command, see *VTAM Operation*.

GWCTL = ONLY|SHR (Interconnection only)

determines whether the host's SSCP does all the gateway control functions during LU−LU session setup. GWCTL determines the SSCP's role only for the gateway NCP being defined and applies only if the SSCP is on the session setup path.

For more information on coding the GWCTL operand, see "The PCCU Definition Statement" on page 287.

INITEST=YES|NO

tells VTAM whether to load the initial test routine: 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 applies only to channel–attached communication controllers. If you code INITEST=YES for an NCP that controls a link–attached communication controller, VTAM ignores it.

LOADSTA=link station name

is an adjacent link station (channel– or link–attached) that is used to load the NCP. Note that 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. (See the notes following the description of DUMPSTA on page 176.)

MAXDATA=size|65535

is the maximum amount of data in bytes, including the transmission header (TH) and the request/response header (RH), that the NCP can receive in one segment of a path information unit (PIU).

MAXDATA should, if possible, be equal to the size of the largest PIU the network will handle. There are restrictions, however. Use these criteria for choosing the value of MAXDATA:

- MAXDATA is related to the BFRS and TRANSFR operands of the BUILD definition statement. MAXDATA should equal the product of BFRS and TRANSFR, minus 18 (BFRS x TRANSFR – 18).
- MAXDATA should not exceed the size of VTAM's MVS VM IOBUF buffer pool; VSE VFBUF buffer pool.
- MAXDATA should not exceed the product of the MAXBFRU and UNITSZ values for the NCP.
- Although MAXDATA applies only to the PCCU statement for a channel–attached NCP, its value should not exceed the capacities of any link–attached NCPs or of any cross–domain NCPs. An SNA path error can occur if MAXDATA exceeds the capacity of any NCP along the route taken by a message.

The default is the maximum size of 65535 bytes.

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 statement for the channel–attachment major node. See "The PU Definition Statement" on page 107 for more information.

MDUMPDS=dumpname

is one of three operands (DUMPDS, CDUMPDS, and MDUMPDS) that name dump files for this communication controller. MDUMPDS applies only to an IBM 3725 Communication Controller.

For more information, see the description of the DUMPDS operand on page 175.

NETID = host network ID (Interconnection only)
is the 1- to 8-character name of the network of the host represented by this PCCU definition statement.

VTAM uses this name in combination with the SUBAREA operand to determine which PCCU definition statement it should process in the NCP major node. Refer to "The PCCU Definition Statement" on page 287 for details.

Do not code this operand if VTAM is started without the NETID start option.

OWNER = ownername
associates a host VTAM with the resources it controls. This operand is used to divide the resources among the hosts that are in session with an NCP. If you code OWNER, physical units (for SNA devices) and lines (for non-SNA devices) can be activated only by the host named in *ownername*. If you do not code OWNER, the resources can be activated by any host, and they are then owned by that host. Then, other hosts cannot activate them.

This operand is closely related to the BACKUP operand, which is described on page 173.

RNAME = link station name|(name,name,....,name)
must match the label on the PU definition statement that describes this NCP in an adjacent communication controller or host. (The PU statement identifies, to the other NCP or host, the link that connects it to this communication controller.) There can be 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.

If you do not supply 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 VARY ACT 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.

Note: For information on how to code RNAME when the NCP is defined in a channel-attachment major node, see the notes following the description of DUMPSTA on page 176.

SUBAREA = n
is the subarea number of the host to which this PCCU definition statement applies. It can range from 1 to 255, and it should match the value assigned to the HOSTSA start option when VTAM was started.

VTAM uses this value to determine which PCCU definition statement it should process in the NCP major node. VTAM examines each PCCU statement until it finds a SUBAREA value that matches its own subarea number. It then uses the associated OWNER name to locate the resources (defined later in the NCP) that it controls.

If you omit SUBAREA, this PCCU definition statement will apply to all hosts that activate this major node and do not find a PCCU statement with a matching SUBAREA value. (In interconnected networks, VTAM also uses the NETID operand to determine which PCCU statement it should process. Refer to "The PCCU Definition Statement" on page 287 for details.)

VFYLM=YES|<u>NO</u>

causes the NCP, when activated, to tell VTAM the name and subarea number of the NCP loaded in the communication controller. It applies only if the VTAM operator requests (or defaults) LOAD=U on the VARY ACT command for an NCP.

When the name and number do not match VTAM's major node name and subarea number for the NCP it is activating:

- If VFYLM=YES, VTAM asks the operator whether it should reload the communication controller with the requested NCP or stop the activation process.
- If VFYLM=NO, VTAM reloads the communication controller with the requested NCP.

When the name matches VTAM's major node name, the value of AUTOSYN applies. (AUTOSYN is described on page 173.)

NCP Definition Statements for the NCP

The definition statements, oriented toward the definition of the NCP load module, and containing VTAM – related information in an NCP major node are the BUILD, SYSCNTRL, HOST, and NETWORK definition statements.

The BUILD Definition Statement

BUILD contains information about how the NCP is to be generated, such as controller storage size, subarea address of the NCP, and characteristics of channel adapters.

For information on defining the BUILD definition statement in a system with SNA network interconnection, see "The BUILD Definition Statement" on page 290.

VTAM places restrictions on these operands of the BUILD definition statement:

COSTAB = class of service table name (Interconnection only)
is a class of service table for the network identified by NETID.

LOADLIB = ddname MVS VM
is 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* when you load the NCP.

MAXSUBA = n (for communicating with V3R1 VM and pre – Version 3 nodes only)
MAXSUBA should be 3 or greater. If coded, it must agree with the value of the MAXSUBA start option and must be in the same range for all nodes in the same network.

If you do not code MAXSUBA, then a pre – Version 3 VTAM cannot function as a gateway SSCP together with this gateway NCP. See "The MAXSUBA Start Option" on page 234 for more information.

NETID = host network ID (Interconnection only)
is the network name of the host as coded on the start option.

Do not code this operand if VTAM is started without the NETID start option.

NEWNAME = symbol
for all NCPs Version 1 Release 3 and later, must be a unique network – wide NCP name. The name must be the same as the name of the NCP major node name.

The 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 (BSC)
RIMM	Reset immediate (BSC)

VTAM needs these OPTION parameters for the listed operator control functions:

NAKLIM	Change line negative polling response limit (MODIFY NEGPOLL)
SESSION	Change session limit (MODIFY SESSION)
SSPAUSE	Change service-seeking pause (MODIFY POLL)
STORDSP	Display NCP storage (DISPLAY NCPSTOR)

The HOST Definition Statement

HOST contains characteristics of VTAM. Knowledge of these characteristics is needed to conduct channel operations for communication with VTAM.

A channel-attached NCP requires one HOST definition statement for each host that will activate the NCP. No HOST definition statement should be defined for a link-attached NCP. For more information, see "Defining Channel-Attached NCP Support" on page 101.

See "The HOST Definition Statement" on page 292 for information on the HOST definition statement in an interconnected network.

The following HOST operands have special VTAM considerations:

INBFRS=nn
defines the number of buffers to be allocated by an NCP for the purpose of receiving data from a host.

MAXBFRU=nn
defines the number of buffers to be allocated by the host for the purpose of receiving data from an NCP.

If you are defining the channel to the NCP in a channel-attachment major node, VTAM may ignore this operand when the NCP is activated by a CMC host. See "Considerations for Defining a Host-to-NCP Channel" on page 101

NETID=host network ID (Interconnection only)
a 1- to 8-character name that identifies the network containing the corresponding host node.

For details on the NETID operand, see "The HOST Definition Statement" on page 292.

UNITSZ=value|64

The UNITSZ value must be the same as the *bufsize* value in the IOBUF buffer pool start option. If more than one NCP is to be active concurrently, the UNITSZ value must be the same for each NCP. The value should be a number of bytes that is evenly divisible by 4. The minimum value of UNITSZ is 64. The maximum value is 4016. The default is 64.

The buffers defined by MAXBFRU and INBFRS will be read from the HOST definition statement and sent to the NCP in the XID format.

The start option buffer pool definitions for IOBUF determine the size of the host buffers to be used for receiving data from an NCP. The size of these buffers will be sent to the NCP in the XID format.

For more information on the use of these options, see *VTAM Customization.*

The LUDRPOOL and PUDRPOOL Definition Statements

VTAM support for switched terminals and dynamic reconfiguration both require that these statements be coded at NCP generation. Support for these functions causes the host to issue Request Network Address Assignment (RNAA) RUs requesting the NCP to assign a PU/LU address to a resource specified by the host. The addresses come from the address pools generated as a result of the LUDRPOOL/PUDRPOOL definition statements. For more information on LUDRPOOL and PUDRPOOL, see *NCP–SSP Resource Definition Reference.*

The NETWORK Definition Statement (Interconnection only)

You can include more than one NETWORK definition statement in the NCP major node to identify each network, other than the native network (which is defined by the NETID operand on the BUILD definition statement) attached to a gateway NCP.

For more information on the NETWORK definition statement, see "The NETWORK Definition Statement" on page 294.

The GWNAU Definition Statement (Interconnection only)

The GWNAU definition statement is an NCP–only definition statement. VTAM does not use the definition statement when processing the NCP generation definitions.

The GWNAU definition statement defines representations of cross–network logical units and SSCPs within the gateway NCP. In some configurations, VTAM's ability to activate SSCP–SSCP sessions depends on proper use of the GWNAU definition statement. These dependencies are discussed in "The GWNAU Definition Statement" on page 295.

NCP Definition Statements for Resources Controlled by the NCP

The NCP generation definition statements that define lines, PU type 4 link stations, and peripheral devices to VTAM are the GROUP, LINE, PU, LU, CLUSTER, and TERMINAL definition statements. Figure 29 on page 185 summarizes the common, VTAM – only, and VTAM – restricted operands for these definition statements. It does not show the conditions (for example, type of line control or type of terminal) under which an operand can be used. For this and other information, consult the individual description of each VTAM – only operand that follows. The information in this book about the VTAM – only operands and restrictions should be used in conjunction with the NCP coding requirements described in *NCP – SSP Resource Definition Reference.*

Use Figure 29 on page 185 to determine what procedure is required to change definition statements and operands for an existing NCP. If VTAM – only operands are to be changed or replaced, no new NCP generation is required. If any NCP definition statement or operand is changed or replaced, a partial or complete NCP generation is required.

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 not all the information coded in the NCP definition statement, you should code all the NCP definition statements with the possible needs of both VTAM and the NCP in mind.

NCP Definition Statements for NCP Resources

Operand	GROUP	LINE	CLUSTER	PU	LU	TERMINAL
ANS				R		
ANSWER	V	V				
AUTODL	R	R				
BHSET	R	R	R			R
CALL	R	R				
DIAL	R					
DISCNT	V	V	V	V		
DLOGMOD	V	V	V	V	V	V
ENCR	V	V		V	V	
FEATUR2	V	V	V	N	N	V
GPOLL			R			
ISTATUS	V	V	V	V	V	V
LINEAUT	R					
LOGAPPL	V	V	V	V	V	V
LOGTAB	V	V	V	V	V	V
LPDATS				N		
MODETAB	V	V	V	V	V	V
OWNER	V	V				
PACING	R	R	R	R	R	
POLIMIT	R	R				
PU	V	V				
SECNET		V		V		
SESSION	R	R				
SSCPFM	V	V		V	V	
TERM	R	R	R	N	N	R
USE	R	R				
USSTAB	V	V	V	V	V	V
VPACING	V	V	V	V	V	
X21SW	R					

Figure 29. NCP Generation Operands Used by VTAM

Legend for Figure 29:

V VTAM – only operand, described in this book.
N VTAM – only operand, described in this book. Refer to *Network Terminal Option Installation* for more information.
R Not a VTAM – only operand; described in the *ACF/NCP – SSP Resource Definition Reference.* VTAM restrictions are described in "Description of VTAM – Only Operands and Restrictions" on page 186.

NCP Definition Statements for NCP Resources

Description of VTAM – Only Operands and Restrictions

ANS = CONTINUE|STOP

determines whether the physical unit continues to operate or stops if the NCP enters automatic network shutdown.

MVS/XA You must code ANS = CONTINUE if the terminal is to participate in XRF sessions.

ANSWER = ON|OFF

is valid only if the SDLC line has dial – in capability. If ANSWER = ON, physical units can dial in to the NCP when the line is active. If ANSWER = OFF, the physical unit cannot dial in to the NCP, regardless of the active – inactive status of the line. The VTAM VARY ANS operator command can change the answer mode.

AUTODL = YES|NO

AUTODL = NO will limit a 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.

BHSET

Block handler sets cannot be dynamically selected for BSC 3270s defined to VTAM. The BHSET operand is permitted for use by other possible owners of the terminal provided that the dynamic BHSET selection option (BHSASSC) is selected on the SYSCNTRL definition statement.

CALL = IN|OUT|INOUT

The CALL operand determines whether terminals, the host processor, or both are able to cause switched connections to be set up over the line to which this operand applies.

If the line is to be used only for terminal – initiated connections, you must code CALL = IN on the LINE statement for the line.

If the line is to be used for connections initiated by terminals, in addition to connections initiated by VTAM, the VTAM operator, or application programs, you must code CALL = INOUT on the LINE statement for the line.

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

There are no VTAM restrictions on CALL = OUT.

Note: CALL = OUT is not supported by non – SNA terminals using the Network Terminal Option (NTO) program product.

DIAL = YES|NO

determines whether the lines in the group require line control procedures. If DIAL = YES, the only operands you may code in the PU statements subordinate to the GROUP statement are MAXLU, PUTYPE, and SPAN.

DISCNT = [YES|NO][,F|NF]

YES|NO

tells VTAM whether to terminate its SSCP – LU and SSCP – PU sessions when the last LU – LU session (involving the logical unit associated with the physical unit) is terminated. The device remains assigned to VTAM and is still active. (That is, sessions can be requested with its LUs; such a request causes PU – SSCP and LU – SSCP sessions to be reestablished. The physical unit itself may also request an SSCP – PU session.)

If you code YES, VTAM is to automatically terminate the SSCP – LU and SSCP – PU session as soon as the last LU – LU session is terminated. If any of the logical units on the physical unit request their own session termination, VTAM ignores the HOLD part of a character – coded logoff or the LAST – NOTLAST part of a field – formatted Terminate Self request. VTAM also rejects any attempt made by the physical unit to terminate its own SSCP session with a Request Discontact request.

If you code NO, VTAM is to terminate the SSCP – LU and SSCP – PU sessions when one of the following conditions is met:

- VTAM receives a normal Request Discontact request from the physical unit, and all the LU – LU sessions associated with that physical unit have been terminated.

- VTAM receives an immediate Request Discontact request from the physical unit. VTAM immediately terminates any existing LU – LU sessions associated with that physical unit. This request overrides any previous normal Request Discontact request from the physical unit.

- All LU – LU sessions have been terminated as a result of:

  - A character – coded logoff request with HOLD = NO,
  - A Terminate Self request for which LAST was specified, or
  - A VTAM VARY INACT command.

  If any LU – LU session was terminated by any other means (such as by an application program CLSDST), the physical unit is not disconnected.

F|NF

tells VTAM whether to indicate “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 VTAM VARY INACT command. If you code F or take the default, “final – use” status is indicated. If you code NF, “not – final – use” status is indicated. 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 = default logon mode table entry name
is the name of the logon mode table entry to be used by default if one is not otherwise provided. If this operand is not coded and the name of a logon mode entry is not otherwise provided, the first entry in the applicable logon mode table (given on the MODETAB operand or defaulted) is used by default.

ENCR = REQD|SEL|OPT|NONE MVS
tells whether this logical unit has any special requirements for enciphering and deciphering messages.

REQD
means that VTAM must encipher all messages to and from this logical unit. If ENCR = REQD is coded, no session can be established with this logical unit unless the host with which it is associated is able to handle cryptographic sessions.

SEL
has no meaning for LUs. If ENCR = SEL is entered, ENCR = OPT is used instead.

OPT
means that this logical unit can engage in cryptographic sessions, but allows the application program to determine whether to use cryptography.

NONE
means that this logical unit cannot engage in cryptographic sessions.

FEATUR2 = (operand,operand...)
describes the machine features for a specific link – attached non – SNA terminal. See *Network Program Products Planning* for a description of start – stop terminal planning considerations. The operands are:

LOWERCSE|DUALCSE
tells VTAM how to send alphabetic characters coded with the TEXT operand on a USSMSG macro instruction to a non – SNA terminal over the SSCP – LU session.

Note: Non – alphabetic characters, as well as all characters coded using the BUFFER operand, are not affected by this operand.

LOWERCSE
means that alphabetic characters will be sent to the terminal over the SSCP – LU session in lowercase.

DUALCSE
means that VTAM will send all characters as they are coded in the USSMSG macro instruction. If you do not code either value, DUALCSE is assumed.

EDATS|NOEDATS
tells VTAM whether this terminal has the extended data stream feature. You cannot use this operand for terminals attached by SDLC lines.

MODEL1|MODEL2
is 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. These characteristics are obtained by using the INQUIRE macro instruction. For more information see, *VTAM Programming.*

PRINTR|NOPRINTR
tells VTAM whether this terminal has an attached IBM 3284 Model 3 printer. This operand is valid only if the TERM=3275 operand is also coded or assumed by NCP definition statement sequencing for this statement.

GPOLL=chars
GPOLL (general polling) must be coded for 3270 non-SNA clusters (that is, 3271 and 3275); VTAM does not support specific polling.

ISTATUS=ACTIVE|INACTIVE
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 LUs, ISTATUS is also effective whenever the associated PU is activated.

This operand "sifts" to subordinate nodes as described in "Sift-Down Effect in VTAM Definition Statements" on page 71.

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 NTO.

LINEAUT=YES
means that VTAM is to consider every LINE in a GROUP as an autocall line. If you omit this operand, but LINEADD=NONE and DIAL=YES, LINEAUT is assumed by default.

LOGAPPL = application program name
is a network-unique name of the application program (1- to 8 alphanumeric characters) to which this logical unit is to be automatically logged on when it is activated. It must correspond to the name assigned to the application program by an APPL or CDRSC definition statement. In cases where a CDRSC may not be defined, the LOGAPPL name must still be unique.

If LOGAPPL is not coded, the terminal operator, an application program, or the domain operator can initiate sessions for this logical unit. In VTAM V3R1.1, you can cancel this automatic logon definition at any time with the VTAM VARY NOLOGON command.

For more information on VARY NOLOGON, and on changing a device's automatic logon definition, see *VTAM Operation*.

LOGTAB = interpret table name
is the member name of an interpret table.

An interpret table can be used to associate a logon request with the name of an application program or routine to which this logon is to be sent. By using an interpret table in conjunction with a USS table, a properly formatted input sequence can be supplied to the USS table. For more information on using interpret tables in conjunction with USS tables, see *VTAM Customization*.

LPDATS = <u>ALLOW</u>|PREVENT
tells whether LPDA is allowed to test a physical unit. The default is ALLOW. This operand is used in dynamic reconfiguration.

MODETAB = logon mode table name
is the name of a logon mode table to be used for the logical unit. Logon mode tables are described in *VTAM Customization*. If this operand is omitted, the IBM-supplied logon mode table is used for the logical unit.

OWNER = ownername|(ownername,...)
is a name used to associate a host VTAM with the resources it controls. This operand is intended for the division of resources in a communication controller. This operand associates this resource or these resources with the PCCU definition statement having the same OWNER value.

PACING = n|0|<u>1</u>
determines how pacing is to be handled between the LU and the NCP to which the logical unit is connected. (In contrast, VPACING involves pacing between VTAM and the NCP.) For more information on pacing, see "Defining Session Pacing Values" on page 76.

n
is the number of normal-flow messages that the NCP is to send to the logical unit before waiting for a pacing response.

n is a decimal integer from 1 through 63, referring to *NCP−SSP Resource Definition Reference* for information on how to select the proper value.

0

means that no pacing is to be performed for sessions with the logical unit.

If the PACING operand is omitted, PACING=1 is assumed by default.

If the logical unit is associated with a PU type 1 (PUTYPE=1), you must either code PACING=1 for the logical unit or omit the PACING operand.

This operand is not valid for statements within a group which has DIAL=YES on the GROUP statement.

POLIMIT=([n|1][,WAIT|NOWAIT|QUEUE])
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. You must code QUEUE.

PU=YES|NO
determines whether a BSC 3270 is to be treated as a physical unit and all terminals subordinate to the cluster are to be treated as logical units. The default value for BSC 3270s is YES. The default value for other non−SNA terminals is NO. If PU=NO (the default), it is assumed that the associated terminals will be supported by a line control program such as the IBM program product Network Terminal Option (NTO) or a similar user−written program. You cannot code this operand on the CLUSTER statement. You must code it on the GROUP or LINE statement, to be sifted down to the CLUSTER level.

SECNET=YES|NO
identifies a physical unit attached to a nonswitched line as being 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 management application.

The default for this operand is the value set for SECNET on the LINE statement, if coded. However, if you do not code SECNET on the LINE statement, its default value on the PU statement is NO.

SESSION=count|1
establishes the NCP line scheduling session limit for the line. It should equal the number of terminals on the line so that it is possible for the NCP to have concurrent sessions with all terminals on the line. This operand refers to NCP polling operations, not SNA sessions.

SSCPFM = FSS|USSSCS|USS3270|USS3275|USS3780|USSNTO
determines whether a logical unit or terminal can support character-coded messages (SSCPFM = USSSCS) in its communication with the SSCP. The default for this operand is SSCPFM = FSS.

Code SSCPFM = USS3270 for terminals attached to a controller defined as an SDLC 3271 (PUTYPE = 1), or SDLC 3275 (PUTYPE = 1). Code SSCPFM = USS3275 to prevent printing of the "Good Morning" message (USS10) for a terminal defined as an SDLC 3275 (PU type 1) that may have an attached printer. For a list of these controllers, see *Network Program Products General Information.*

Code SSCPFM = FSS or SSCPFM = USSSCS for all other terminals for which the SSCPFM operand is valid. Consult the individual terminal component description manual to determine whether character-coded messages (SSCPFM = USSSCS) or formatted messages (SSCPFM = FSS) are supported for SSCP communication.

Code SSCPFM = USS3780 for non-SNA devices supported with the Network Terminal Option (NTO) program product that use the 3780 protocol (see *NTO Installation*). These devices use USS command facilities (LOGON and IBMTEST) and are supported only by Release 2 of NTO.

Code SSCPFM = USSNTO for all other non-SNA devices supported with NTO. These devices use the USS command facilities and are supported by all releases of NTO.

TERM = type
describes, for a virtual logical unit supported through NTO, the device data stream compatible characteristics. These characteristics are placed in an 8-byte program-supplied storage area, after an INQUIRE DEVCHAR macro instruction is executed by an application program that specifies the virtual logical unit name. In particular, this information will be in the DEVAUXTP field of the storage area that is mapped by the DEVCHAR DSECT. For more information on the DEVCHAR DSECT, see *VTAM Programming.*

USE = NCP|EP
USE = EP prevents the line from being activated unless a VTAM VARY ACT command is issued for it. Activating such a line with a VTAM VARY command gives the line to VTAM and NCP whether it was in use by EP.

USSTAB = USS definition table name
is the name of a USS definition table to be used for the logical unit. USS definition tables are described in *VTAM Customization.* If this operand is omitted, the IBM-supplied USS definition table is used for the logical unit when character-coded requests are received.

If USSTAB is to be coded on a CLUSTER or TERMINAL statement, you must also code PU = YES for that statement.

VPACING = n|0|2

determines whether pacing is to be performed (during a session) with the logical units represented by the appropriate NCP definition statements. For more information on pacing, refer to "Defining Session Pacing Values" on page 76.

n

is the number of data requests that VTAM is to send to an NCP (for an associated session) before waiting for a pacing response. No further requests can be sent to the NCP until the NCP replies with a pacing response to VTAM.

0

means that no pacing is to be performed.

If the VPACING operand is omitted, VPACING = 2 is assumed.

This operand is not valid for statements within a group which has DIAL = YES on the GROUP statement.

X21SW = YES|NO

X21SW = YES means that all 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. The default is NO.

Dynamic Reconfiguration Statements

The first definition statement is VBUILD TYPE=DR. To code a DR member, use ADD and DELETE statements, each followed by NCP generation PU and LU statements as required. The completed set of statements must obey the rules described in "Format of Definition Statements in this Book" on page 74, and they must be filed in the VTAM definition library. For an example of statements defining a dynamic reconfiguration file, see Figure 30.

The formats of these statements are:

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=DR
[name]	ADD	TO=resource name
[name]	DELETE	FROM=resource name

You can code any VTAM-only operands in a DR deck.

```
[name]      VBUILD  TYPE=DR
            DELETE  FROM=LINE4
PU3791A     PU      (You need not code operands
PU3791B     PU      and associated LUs)
            DELETE  FROM=PU3271A
LU3277B     LU
LU3277C     LU      (operands not required)
LU3274D     LU
            ADD     TO=LINE4
PU3791C     PU      (Code operands you would code in normal
LU3790A     LU      network definition for these PUs and LUs.)
LU3790B     LU
            ADD     TO=PU3271A
LU3274E     LU
LU3274F     LU      (Code operands you would code in normal
LU3274G     LU      network definition for these LUs.)
LU3277H     LU
LU3274I     LU
```

Figure 30. An Example of a Set of Dynamic Reconfiguration Statements

- In a PU statement:
  - ADDR
  - ANS
  - BNNSUP (only valid when PUTYPE=1 and the device is an SDLC 3270, for example, a 3271 Model 12)
  - IRETRY
  - LPDATS
  - MAXDATA
  - MAXOUT
  - PASSLIM
  - PUTYPE (either 1 or 2).

You must include MAXDATA on all PU statements in a DR deck; MAXDATA does not default.

- In an LU statement:

  - BATCH
  - LOCADDR
  - PACING.

  You must include LOCADDR on all LU statements in a DR deck; LOCADDR does not default.

For any other NCP-related operand, the default is used.

Sifting takes place within the hierarchy of minor nodes being added dynamically. However, 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 DR deck statements for the resources being added.

name
name is optional. If you code the name for a DR statement, VTAM uses it to refer to the set of changes in operator messages.

TYPE=DR
means that this VBUILD statement defines a dynamic reconfiguration file to VTAM. This operand is required.

resource-name
names either a nonswitched line for adding and deleting physical units or a PU type 1 or PU type 2 for adding and deleting logical units.

The PU and LU statements are the same as the statements ordinarily defining these resources. For example, suppose a user wishes to move PU6 and its associated logical units (LU1 and LU2) from LINE6, where it is currently attached, to LINE7 to see if the logical units' response time can be improved. First, a DRDS file must be defined.

Whenever a physical unit is deleted, all associated logical units are also deleted.

You do not need to code operands in the PU and LU statements for resources that are being deleted.

You must code operands on the PU and LU statements in the same format as they appear in the NCP major node. Moreover, you must code all desired operands, as there is no "sift-down" from the GROUP, LINE, or PU to which the physical unit or logical unit is being added.

For efficient storage management, DELETEs should be processed before ADDs.

For example, if only the logical unit LU1 is to be moved from PU3 to PU4, the dynamic reconfiguration file could be coded as shown in Figure 31.

```
          VBUILD    TYPE=DR

[STEPA]   DELETE    FROM=PU3
LU1       LU        operands,...(not recommended—ignored)
[STEPB]   ADD       TO=PU4
LU1       LU        operands,...
```

Figure 31. Dynamic Reconfiguration Statements for Moving an LU to another PU

The names on the ADD and DELETE statements are optional; however, if you code them they are included in error messages issued to the VTAM operator if the ADD or DELETE subsequently fails. In the preceding example, if you attempted to delete LU1 from PU3 while LU1 was still active, STEPA would fail and "STEPA" would be included in the error message. Note that if a step fails, any subsequent steps that depend on the failing step will also fail. In the preceding example, if STEPA fails to delete LU1, STEPB will also fail because you are attempting to add LU1 to PU4 while the LU1 is already associated with PU3 in the current network configuration.

Defining VTAM Routes

PATH definition statements are representations of the routes VTAM takes to communicate with other subarea nodes. One or more PATH statements are filed in the VTAM definition library. The name assigned to this path definition set is used to activate this set of paths. More than one path definition set may be filed and activated. Note that the VBUILD statement is not used.

Communication with other subarea nodes is not possible unless all the required path tables are active. It is recommended, therefore, that any required path tables be activated before any NCP or channel–attachment major nodes are activated. See *Network Program Products Planning* for an illustration of how PATH definition statements fit into the VTAM routing structure.

VTAM Route: PATH

The PATH Definition Statement

Code the PATH statement like this:

Name	Definition Statement	Operands
name	PATH	DESTSA=n|(n1,n2,n3,...) [,ER0=(adjsub[,tg#])] [,ER1=(adjsub[,tg#])] . . . [,ER7=(adjsub[,tg#])] [,VR0=er#] [,VR1=er#] . . . [,VR7=er#] [,VRPWS00=(min#,max#)] [,VRPWS01=(min#,max#)] [,VRPWS02=(min#,max#)] [,VRPWS10=(min#,max#)] [,VRPWS11=(min#,max#)] [,VRPWS12=(min#,max#)] [,VRPWS20=(min#,max#)] . . . [,VRPWS72=(min#,max#)]

name

provides a name for the PATH statement and is not checked by VTAM for validity. Its value is to point out the particular PATH statement in case of a definition error or warning message during activation.

DESTSA=n|(n1,n2,n3,...)

n is the destination subarea number for the routes being defined to subarea nodes in your network. You may code more than one destination subarea as long as the route definition statement holds for each one.

ER0=(adjsub[,tg#]),...,ER7=(adjsub[,tg#])

names the adjacent subarea and, optionally, the transmission group number for the associated explicit routes originating in the host and leading to the destination subarea.

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.

tg#

is an optional transmission group number for the explicit route being defined. VTAM always uses the value 1 for channel links.

VR0 = er#,...,VR7 = er#
associates the virtual route with the explicit route.

er#
is the explicit route number to which the virtual route is mapped as a decimal integer between 0 and 7.

Note: You must code at least one of the keyword identifiers for routes (ER0,...,ER7, VR0,...,VR7).

VRPWS00 = (min#, max#), . . ., VRPWS72 = (min#, max#)
is the default 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.

If you do not code default values, 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 these 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, see *Network Program Products Planning.*

Defining a Multiple – Domain VTAM Network

This section describes how to define CDRMs, CDRSCs and default SSCP lists within an adjacent SSCP table. Additional path statements are required for cross – domain communication. For more information, see "Defining VTAM Routes" on page 197.

Defining Cross – Domain Resource Managers

One or more major nodes can be defined for cross – domain resource managers (CDRMs). Each CDRM major node is defined with a VBUILD statement and each minor node is defined with a CDRM definition statement. Within a domain there must be exactly one cross – domain resource manager defined for every other domain (in this network or another network) with which this domain will communicate.

Note: Each domain must also have a definition of its own cross – domain resource manager.

You can use the CDRM definition statement to define resources in other networks. See "The CDRM Definition Statement" on page 253 for details.

Cross – Domain Resource Manager: VBUILD

The VBUILD Statement

Code a VBUILD statement for each set of CDRM definition statements.

Code the VBUILD statement like this:

Name	Definition Statement	Operands
[name]	VBUILD	TYPE = CDRM [,CONFGDS = name] [,CONFGPW = password]

name
name is optional.

TYPE = CDRM
means that the VBUILD statement defines a CDRM major node to VTAM. This operand is required.

CONFGDS = name
is a 1 – to 8 – character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the VTAM start procedure. Refer to *Network Program Products Planning* for a description of configuration restart.

VM This operand is ignored if coded on a VM system.

CONFGPW = password
is the 1 – to 7 – character alphanumeric password, if required, for VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM prompts for the correct password when VTAM attempts to open the data set. Use this operand only when you also use CONFGDS.

VM This operand is ignored if coded on a VM system.

The NETWORK Definition Statement (Interconnection only)

The NETWORK definition statement names the network in which each of the CDRMs being defined resides. The NETWORK statement appears after the VBUILD statement and immediately before one or more CDRM statements defining CDRMs residing in the network.

Name	Definition Statement	Operands
[name]	NETWORK	[NETID = network id]

name

is a 1 – to 8 – character name that identifies this NETWORK 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.

NETID = network id

is an optional 1 – to 8 – character name that identifies the network in which the CDRM or set of CDRMs reside. This NETID pertains to the CDRM or set of CDRMs defined following the NETWORK statement and before any other NETWORK statement in the same set of definitions.

For more information on coding the NETWORK definition statement, see "The NETWORK Definition Statement" on page 252.

Cross – Domain Resource Manager: CDRM

The CDRM Definition Statement

CDRM definition statements define cross – domain resource manager (CDRM) minor nodes. One or more of these statements, when preceded by a VBUILD statement, define a CDRM major node.

Name	Definition Statement	Operands
cdrmname	CDRM	[CDRDYN = YES|**NO**] [,CDRSC = OPT|**REQ**] [,ELEMENT = n|**1**] [,ISTATUS = **ACTIVE**|INACTIVE] [,RECOVERY = **YES**|NO] [,SPAN = (spanname)] [,SUBAREA = n] [,VPACING = n|0|**63**]

cdrmname
is the minor node name of the CDRM represented by this statement.

For special considerations for cross – network resources, see "Defining Cross – Domain Resources" on page 208.

CDRDYN = YES|NO
determines whether a CDRM is authorized to dynamically define CDRSC representations of cross – domain or cross – network resources at the time a session request is received from or sent to the CDRM that manages the resource.

YES
authorizes dynamic definition of cross – domain or cross – network resources by this CDRM, eliminating the need to predefine certain CDRSCs. It also authorizes adjacent SSCP trial and error routing. Dynamic definition is subject to the CDRSC operand value on the CDRM sending the Initiate request. If you are defining a gateway SSCP, you *must* code CDRDYN = YES.

Note: VM In VTAM V3R1, CDRDYN = YES works differently in VM. In MVS and VSE, and in VTAM V3R1.1 for VM, both origin LUs and destination LUs can dynamically build definitions for cross – domain resources. In V3R1 for VM, only origin LUs can do so.

NO
causes the failure of any cross – domain or cross – network session request in behalf of a cross – domain or cross – network resource that has no predefined CDRSC entry. NO is the default.

Note: Use this operand with the CDRSC operand to set up dynamic CDRSC definition. See Network Program Products Planning for more information on this function.

CDRSC = OPT|REQ

determines whether resources owned by the CDRM being defined by this statement may be dynamically defined at the time a session request is received from or sent to this external CDRM which manages the resource. This allows dynamic CDRSC definition to be performed only when the cross – domain or cross – network session requests are received from or sent to certain domains.

OPT

authorizes dynamic definition of cross – domain or cross – network resources owned by this CDRM, eliminating the need to predefine certain CDRSCs. VTAM builds control blocks for these resources at CDINIT time.

REQ

causes the failure of any cross – domain or cross – network request in behalf of a cross – domain resource that has no predefined CDRSC entry.

Note: Use this operand with the CDRDYN operand to set up dynamic CDRSC definition. See Network Program Products Planning for more information on the dynamic CDRSC definition function.

ELEMENT = n|1

is the element part of the network address of the CDRM. This operand, in conjunction with the SUBAREA value, defines the network address of the CDRM. For the CDRM definition that represents a host VTAM, the ELEMENT value must be 1. For external CDRMs (CDRMs in other domains or networks) this number can be a decimal integer ranging from 0 to 32767. If you do not code ELEMENT, ELEMENT = 1 is assumed by default. For SNA network interconnection considerations, see "Defining Cross – Domain Resource Managers" on page 251.

ISTATUS = ACTIVE|INACTIVE

is the initial status of the CDRM.

For the host CDRM, ACTIVE means that it is capable of going into session with an external CDRM.

For the host CDRM, INACTIVE means that it is *not* capable of going into session with an external CDRM.

For an external CDRM, ACTIVE means that a CDRM – CDRM session should be established from the CDRM in the host in which this set is being defined to the CDRM named by this statement. Note that the attempt to establish this session fails if the host CDRM is not active, so you should code ISTATUS = ACTIVE only if:

- This CDRM is in the same major node as the host CDRM (for which ISTATUS = ACTIVE as well), or
- This CDRM major node is to be activated after the host CDRM is active.

For an external CDRM, INACTIVE means that a CDRM – CDRM session is *not* to be established.

After its major node has been activated, a CDRM defined as INACTIVE can be activated by using the VARY command, or in the case of a CDRM in another domain, by receiving a session – activation request from that domain.

RECOVERY = YES|NO
determines whether the SSCP – SSCP session between the named CDRM and the host CDRM should be restarted automatically when a session outage occurs.

Both the host CDRM and the external CDRM *must* have RECOVERY = YES (the default) to enable automatic recovery.

VM This operand is ignored if coded in a V3R1 VM system.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

SUBAREA = n
is the number of the subarea in which the CDRM represented by this statement resides. This number can range from 1 to 255. The SUBAREA value for each subarea to be used in cross – domain or cross – network communications must be unique throughout the network, but not throughout a set of interconnected networks. For a same – domain CDRM, this SUBAREA value must be the same as the host's subarea. This operand is required for cross – domain communication, but it is optional for cross – network communication.

See "Defining Cross – Domain Resource Managers" on page 251 for SNA network interconnection considerations concerning the SUBAREA operand.

VPACING = n|0|63
is the maximum number (in decimal) of requests that other CDRMs should send to this CDRM before waiting to receive a pacing response. This value is exchanged during CDRM – CDRM session establishment. If the value is not between 0 and 63, the maximum of 63 is used. If VPACING = 0, no pacing is done.

The GWPATH Definition Statement (Interconnection only)

Each GWPATH definition statement defines a possible cross-network session path between the VTAM host CDRM and a CDRM in another network and their related LU-LU sessions.

The GWPATH definition statement is optional. You may code one only if VTAM is started with a NETID start option that identifies the network to VTAM. The format of the GWPATH statement is:

Name	Definition Statement	Operands
[name]	GWPATH	[ADJNET=adjacent network ID] [,ADJNETEL=adjacent network element|1] [,ADJNETSA=adjacent network subarea] [,ELEMENT=element address|1] [,GWN=gateway NCP name] [,SUBAREA=subarea address]

For more information on the GWPATH definition statement and on the operands, see "The GWPATH Definition Statement" on page 255.

Cross – Domain Resources

Defining Cross – Domain Resources

One or more major nodes can be defined for cross – domain resources (CDRSCs). Each CDRSC major node is defined with a VBUILD definition statement and each minor node is defined with a CDRSC definition statement.

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 will not be 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 wishes to relinquish ownership of the logical unit, it releases or deactivates the logical unit and its associated physical unit, causing VTAM to activate its CDRSC representation—again making it a cross – domain resource.

Note that the process of relinquishing ownership *does* disrupt ongoing sessions. For further information on this function, refer to *Network Program Products Planning.*

The VBUILD Statement

Code a VBUILD statement for each set of CDRSC definition statements.

Code the VBUILD statement like this:

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=CDRSC [,CONFGDS=name] [,CONFGPW=password]

name
name is optional.

TYPE=CDRSC
means that this VBUILD statement defines a CDRSC major node to VTAM. This operand is required.

CONFGDS=name
is a 1– to 8–character data definition name that identifies the configuration restart data set defined by the user for this major node. Include a DD statement using this data definition name in the VTAM start procedure. Refer to *Network Program Products Planning* for a description of configuration restart.

VM This operand is ignored if coded on a VM system.

CONFGPW=password
is the 1– to 7–alphanumeric password, if required, for VTAM to gain access to the configuration restart data set. If you do not code CONFGPW but it is required by VSAM, VSAM prompts for the correct password when VTAM attempts to open the data set. Code this operand only if you also code CONFIGDS.

VM This operand is ignored if coded on a VM system.

Cross-Domain Resource: NETWORK

The NETWORK Definition Statement (Interconnection only)

The NETWORK definition statement gives the name of the network in which the CDRSC resides. A NETWORK statement, followed by one or more CDRSC statements associated with that network, is used to define cross-network resources in the CDRSC major node.

Name	Definition Statement	Operands
[name]	NETWORK	[NETID=network id]

name
is a 1- to 8-character name that identifies this 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.

NETID = network id
is an optional 1- to 8-character name of the network in which the set of CDRSCs reside. NETID must be unique within your configuration.

For more information on coding the NETWORK definition statement, see "The NETWORK Definition Statement" on page 272.

The CDRSC Definition Statement

CDRSC definition statements define cross – domain resource (CDRSC) minor nodes for those cross – domain resources with which an application program or logical unit within this domain can have a session. One or more of these statements, when preceded by a VBUILD statement, defines a CDRSC major node. These are optional in some cases; see *Network Program Products Planning* for more information.

Notes:

1. *Information on coding the CDRSC statement for TSO/VTAM is in "Multiple – Domain Network" on page 327.*
2. *The same set of CDRSC statements can be used throughout the network.*
3. *You can use the CDRSC definition statement to define resources in other networks. For details, see "The CDRSC Definition Statement" on page 273.*

Name	Definition Statement	Operands
cdrscname	CDRSC	[CDRM = cdrmname] [,ISTATUS = **ACTIVE**|INACTIVE] [,SPAN = (spanname)]

cdrscname
is the name of a resource in another domain or network (a logical unit or application program) represented by this statement and is required. If the resource is a VTAM application program, the name must be the same as the one in the "name" field (not the ACBNAME operand) of the application program's APPL statement in the VTAM definition in the other domain.

For special considerations when the CDRSC definition statement defines a resource in another network, see "The CDRSC Definition Statement" on page 273. Network Interconnection Considerations."

CDRM = cdrmname
names the CDRM that is in the same domain as the cross – domain resource, and which controls the resource.

VM You must code this operand in VTAM V3R1 for VM; otherwise it is optional.

When you do not code the CDRM operand, the default SSCP list is used to send session setup requests to SSCPs in the order in which they appear in the list, until the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM updates the static CDRSC with its name.

When the name coded in the CDRM operand is also in the default SSCP list, this CDRM name is moved to the top of the list. When the name coded in the CDRM operand is not in the default SSCP list, VTAM places the CDRM name at the top of the list. The default SSCP list is then used to send session setup requests to SSCPs in the order in which they appear in the list, until the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM updates the static CDRSC with its name. This action eliminates the need for the network operator to issue a MODIFY CDRM command because the owning SSCP is different from the CDRM name.

ISTATUS = ACTIVE|INACTIVE
is the initial status of this CDRSC. Logically active means that the session – establishment requests can be sent to the named CDRM.

ISTATUS = ACTIVE means that this resource is logically active to this domain. The resource is not necessarily active in its own domain.

ISTATUS = INACTIVE means that this resource is not logically active to this domain. The resource is not necessarily inactive in its own domain.

SPAN = (spanname)
Code this operand if you are using NCCF or NetView. For a full description, refer to either *NCCF Installation and Resource Definition* or the *NetView Installation and Administration Guide.*

Defining Default SSCP Lists Within an Adjacent SSCP Table

The default SSCP list is similar to the adjacent SSCP table that is used in SNA network interconnection. The default SSCP list is a list of SSCPs that may be in session with the VTAM SSCP. When the VTAM SSCP is in session with these SSCPs, session-initiation requests (for LU-LU sessions in which the requested session partner is not in VTAM's domain) may be routed to these SSCPs, one at a time, until the owner of the requested resource is located or the end of the list is reached. One or more adjacent SSCP tables may be activated in each VTAM host. The definitions are filed in a VTAMLST file.

A default SSCP list for a single network consists of a VBUILD statement with TYPE=ADJSSCP followed by a set of ADJCDRM definition statements.

This section describes how to use default SSCP selection in a single network. For information on defining adjacent SSCP tables for interconnected networks, see "Defining Adjacent SSCP Tables" on page 275.

VM In V3R1 for VM, this function is not available.

Default SSCP List: VBUILD

The VBUILD Statement

Code a VBUILD statement for each default SSCP list.

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=ADJSSCP

name
The *name* is optional.

The ADJCDRM Definition Statement

Consecutive ADJCDRM definition statements comprise the list of default SSCPs.

Name	Definition Statement	Operands
cdrmname	ADJCDRM	

cdrmname
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 statement must have a unique cdrmname.

There are no operands on the ADJCDRM definition statement.

Alternatives to Predefining Cross – Domain Destination Logical Units

There are three ways to lay the groundwork for communicating with a destination LU in another domain. One is to define the destination logical unit to VTAM as a cross – domain resource owned by an external CDRM, as described in "Defining Cross – Domain Resources" on page 208. However, you can also use either default SSCP selection or the alias name translation facility to allow communication with a cross – domain destination logical unit. These alternatives are discussed in the following sections.

MVS VSE Default SSCP Selection

If you have defined a default SSCP list, VTAM sends the session setup request for an undefined destination logical unit to SSCPs in the order in which they appear in the list, until the owning SSCP is found or the end of the list is reached. If there is no SSCP – SSCP session with an SSCP named in the list, the request is ignored.

If the SSCP to which a same – network session setup request is routed does not own the destination logical unit, the request is rejected.

During this trial – and – error routing of a cross – domain initiate (CDINIT) request, the request may be routed to an SSCP that also has a default SSCP list. If so, that SSCP will not reroute the request because it resides in the same network (that is, it has the same NETID). Once the owning SSCP is found, VTAM automatically creates a CDRSC for the logical unit, if dynamic definition of cross – domain resources is allowed.

Using default SSCP selection means you do not have to code cross – domain resource (CDRSC) definition statements for logical units in other domains, but VTAM's performance is slower because of the time it takes to send startup requests to SSCPs that do not own the logical unit.

Refer to "Defining Default SSCP Lists Within an Adjacent SSCP Table" on page 213 for more information about defining a default SSCP list.

Note: When an invalid resource name is entered in a logon request, the request is not rejected until every SSCP in the default SSCP list has been sent the initiate request, in turn, and has rejected it.

MVS VM Using an Alias Name Translation Facility

Another alternative to predefining cross-domain destination logical units to VTAM is to:

- Use the alias name translation facility.
- Assign a NETID for the network when starting VTAM (see "MVS VM The NETID Start Option (Interconnection only)" on page 235).
- Define all logical units (or, at least, those that will have cross-domain sessions) in the network to the name translation facility.

This alias name translation function was developed for resolving duplicate logical unit names when interconnecting two or more independent networks. However, in a system with one network, this function can be used to determine the owning SSCP for a requested resource.

To use this method, you must define all the logical units in the network and their owning SSCPs to the name translation facility. Define each alias name to be the same as the real name. If VTAM does not have a definition of a destination logical unit for a session setup request, it calls the name translation facility with the name of the logical unit. The name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network ID and the logical unit's alias name in that network. Since this system has just one network, the network ID is VTAM's network ID; and the alias name is the same as the real name. Thus, the name translation facility is used as a directory to the owning SSCP.

If the SSCP has a session with the destination logical unit's SSCP, and dynamic definition of cross-domain resources is allowed, the CDINIT request for the session is sent to the destination logical unit's SSCP. VTAM automatically creates a CDRSC definition table entry for the logical unit. (You can allow dynamic definition of a resource by coding CDRDYN=YES on the CDRM statement for the VTAM host CDRM and CDRSC=OPT on the CDRM statement for the destination logical unit's SSCP.) Refer to "Using an Alias Name Translation Facility" on page 269 for information on how to define a name translation facility to VTAM.

If you are not using an alias name translation facility or if the name translation facility fails to return the name of the destination logical unit's owning SSCP, default SSCP selection is tried next. See the preceding section for a discussion of this topic.

Chapter 6. Defining Start Options

If you are already familiar with the VTAM start options, refer to the alphabetic listing in Appendix B, "Quick Reference for VTAM Start Options" on page 323.

If you are unfamiliar with the start options, refer to this chapter for explanations.

This chapter discusses:

- Start option sources
- Coding start options
- Creating start option and configuration lists.

Start Option Sources

When VTAM is started, options can be provided from any combination of the following sources:

- IBM-supplied values internal to VTAM. These are used as values for options that are not provided in one of the ways listed below. These values are the defaults shown for the options in "Coding Start Options" on page 220. Note that the SSCPID is not one of the IBM-supplied values; you must code it when starting VTAM.

- Default start option list. When VTAM is started, it reads from the VTAM definition library a list of user-defined default values named ATCSTR00.

- Supplemental start option lists. Among the start options entered by the VTAM operator is the LIST start option, which can be entered at the console. Using the LIST start option, you can name another list to supplement ATCSTR00. This list can be entered when prompted and may override options for any option in the list, as well as for options not in the list.

- Prompted start options. You can code PROMPT only in ATCSTR00 to request that the operator be prompted to enter further start options.

- Start options reentered by the system operator to correct errors in the original entry. If VTAM detects an error in a start option list, it notifies the system operator, who must reenter that option.

How Options Can Be Overridden

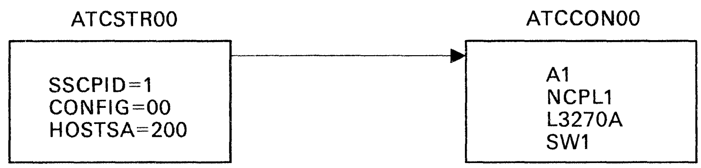
An option from one source can override the same option from another source, depending on the relative priority of the sources. The list of start option sources above is arranged from the lowest priority to the highest. For example, if ATCSTR00 has CONFIG=01, but the operator enters LIST=ST and ATCSTRST contains the option CONFIG=02, the configuration defined by ATCCON02 is activated.

Any or all of the start options can be filed together in a single list. As noted above, the default start option list (ATCSTR00) is automatically processed when VTAM is started; therefore it must exist before VTAM is started.

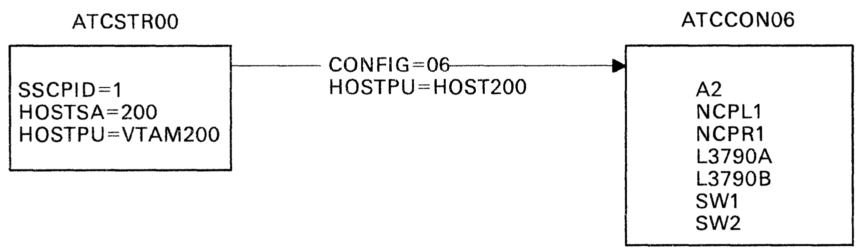
Figure 32 on page 219 contains examples of the start procedure that shows how you can use CONFIG and LIST.

Refer to "Creating Start Option and Configuration Lists" on page 221. VM For more information on customizing start option lists, see *VTAM Customization.*

Example 1. The operator starts VTAM with no start options. The automatically processed member ATCSTROO contains a set of start options including CONFIG=00. The result is:



Example 2. The operator starts VTAM with the following start options: CONFIG=06 and HOSTPU=HOST200. CONFIG=06 refers to ATCCONO6, which contains a list of major node names. The HOSTPU option overrides the option filed in ATCSTROO. The result is:



Example 3. The operator starts VTAM with the start option LIST=ST. LIST=ST refers to ATCSTRST, which contains a set of start options. The same configuration and start options are obtained as in Example 2. The result is:

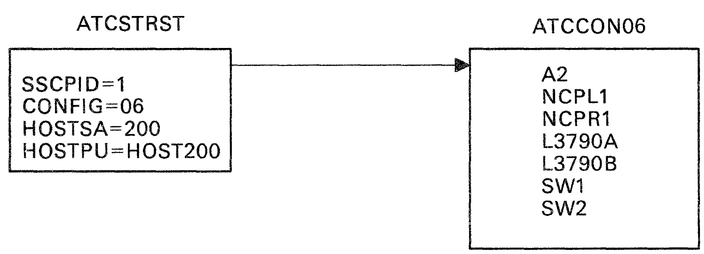


Figure 32. Examples of Using CONFIG and LIST Start Options

Coding Start Options

The VTAM operator can enter start options (unless otherwise noted in the following list of start options) as operands on the START command, when prompted during the start procedure, or by using a start option list. If the VTAM operator enters the start options as operands on the START command, the length of the entire number of options cannot exceed the console's line length. Refer to "Overriding Buffer Pool Values" on page 227.

Note: When entering start options from the console due to a prompt by VTAM, if all of the desired options cannot be entered on one line, place a comma after the last option on the line of input and VTAM will prompt you for more start options.

MVS The START Command

To start VTAM in MVS, use the START command:

```
START procname,,,(option,option,...,option)
```

where:

procname
is the name of the VTAM start procedure.

option
is one of the VTAM start options. The total number entered is limited by the line length of the console.

VSE The EXEC Command

To start VTAM in VSE, use the system EXEC command:

```
EXEC PROC=procname
```

where:

procname
is the name of the VTAM start procedure.

The start options are contained in the file ATCSTR00. Inside ATCSTR00, you can determine whether the system operator is prompted for more start options every time he or she starts VTAM.

VM The VTAM START Command

To start VTAM in VM, use the VTAM START command:

VTAM START (option,option,...,option)

where:

option
is one of the VTAM start options. The total number entered is limited by the line length of the console.

Creating Start Option and Configuration Lists

If a start option list is to be used, you must create a list and put it in a VTAMLST file named ATCSTRyy (MVS a member named ATCSTRyy in SYS1.VTAMLST). You must also file a list named ATCSTR00. If no start options are to be filed in ATCSTR00, file a record of blanks to avoid receiving a warning message.

Notes:

1. VM *Whenever you change start option lists, you must use the GCS ACCESS command to re−access the disk that contains the corresponding VTAMLST files before you restart VTAM.*

2. VM *Start options for VSCS are coded as operands on the DTIGEN macro. The macro and its operands are described in "The DTIGEN Macro Instruction" on page 341.*

Whether or not the VTAM operator enters LIST=yy as a start option, VTAM attempts to locate ATCSTR00. If it does not exist, VTAM sends an error message to the VTAM operator, followed by a message prompting the operator for start options.

If a configuration list is specified either in a start option list or by the VTAM operator, this optional list of major node names must have been created and put in the VTAM definition library under the file name (or member name) ATCCONxx. The same major node name can be in more than one configuration list. For example, a group of channel−attached terminals named LOCCON01 can be in ATCCON00 and ATCCON06. However, LOCCON01 can only be activated once during VTAM initialization.

The major node names in the configuration list correspond to the names assigned to the major nodes when they are defined and put into the VTAM definition library using the system utility program. If no configuration lists exist, the VTAM operator must use the VARY command to activate each major node.

VTAM definition statements can be filed during the same execution of IEBUPDTE used for filing start options.

Formats

Code these lists as one or more 80 – byte records in this format:

card column 1	**card column 71**
[/...]item[,item]...	...

[/...] signifies that one or more blanks may precede the first item.

item represents (1) a major node name, if a configuration list is being coded, or (2) a start option, if a list of start options is being coded.

Separate items with commas. There must be no intervening blanks.

Column 72 is reserved for the continuation indicator. VTAM ignores columns 73 through 80.

To continue onto another record, place a comma after the last item in the current record, enter a nonblank character in column 72 of the current record, and start the next item to the right of column 16 in the new record. If desired, the current record can be coded through column 71, a nonblank character entered in column 72, and the item continued in any column of the new record.

To create a comment card, enter an asterisk (*) in column 1 of any card other than a continuation card.

Start Option Formats

The formats of the start options shown in this chapter follow the same conventions as those used for definition statements in Chapter 5, "Defining the Network to VTAM." These start options are as follows:

SSCPID=n
[,poolname=(baseno,bufsize,slowpt,F,xpanno,xpanpt)][1]
[,CDRSCTI=n|**480**]
[,**COLD**|WARM]
[,CONFIG=xx|**00**|name]
[,CSALIMIT=**0**|n|nK|nM] MVS VM
[,CSA24=**0**|n|nK|nM] MVS/XA
[,DLRTCB=n|**32**][2] MVS VM
[,HOSTPU=host subarea PU name|**ISTPUS**][3]
[,HOSTSA=n|**1**]
[,IOINT=n|**180**]
[,ITLIM=n|**0**]
[,LIST=yy]
[,MAXAPPL=n|**10**] VM *V3R1 only*
[,MAXSUBA=n|**15**]
[,MSGMOD=YES|**NO**]
[,NETID=network id] *Interconnection only*
[,NODELST=name]
[,PPOLOG=YES|**NO**] (V3R1.1 only)
[,**PROMPT**|NOPROMPT][4]
[,SONLIM=([m|**60**][,t|**30**])]
[,SSCPDYN=**YES**|NO]
[,SSCPNAME=name] *Interconnection only*
[,SSCPORD=**PRIORITY**|DEFINED]
[,SUPP=**NOSUP**|INFO|WARN|NORM|SER]
[,TNSTAT[,CNSL|**NOCNSL**][,TIME=n|**60**]|**NOTNSTAT**]
[,TRACE|**NOTRACE**,ID=nodename,TYPE=BUF|IO[,EVERY]][3]
[,TRACE|**NOTRACE**,ID=linename,TYPE=LINE[,COUNT=n|**ALL**]][3]
[,TRACE|**NOTRACE**,ID=nodename,TYPE=SIT[,COUNT=n|**ALL**]][3]
[,TRACE|**NOTRACE**,ID=VTAMBUF,TYPE=SMS][5] [6]
[,TRACE,TYPE=VTAM
    [,MODE=**INT**|EXT]
    [,OPTION=ALL|option|(option,option,...,option)][7]
    [,SIZE=n|**2**]]
[,**NOTRACE**,TYPE=VTAM][8]
[,USSTAB=tablename]
[,VTAMEAS=n|**404**] VM *V3R1 only*

Notes:

1. *You can code options for more than one buffer pool.*

2. VM *In V3R1, the default for DLRTCB is 8.*

3. VM *HOSTPU does not apply in V3R1.*

4. *The VTAM operator cannot enter the PROMPT|NOPROMPT start option. It can only be coded in a predefined list.*

5. *Do not use NOTRACE when starting VTAM, except to override a TRACE start option coded in a predefined list. Also, VTAM accepts more than one TRACE start option during initialization. You can code only one type of VTAM trace in each TRACE|NOTRACE start option (whether selected by the VTAM operator or predefined in a list of start options).*

6. *For TYPE=SMS, the only valid value for ID is VTAMBUF.*

7. *You can use the abbreviation OPT for the OPTION operand.*

8. VM *In V3R1, TRACE is the default instead of NOTRACE.*

The SSCPID Start Parameter

SSCPID=n
is a decimal integer from 0 to 65535. It is part of an SSCP identifier used when a PU or external CDRM establishes contact with VTAM. The SSCPID must be unique within a network. In addition, the SSCPID in the VTAM host being started must be different from those of SSCPs in other networks that may be in session with this SSCP. See "The SSCPID Start Parameter" on page 296 for SNA network interconnection considerations. This parameter is required for both same- and cross-domain sessions.

VTAM uses the SSCPID value to construct a 48-bit identification sequence that is sent to a PU when a session is established using the SNA ACTPU command. In a multiple-domain network, it is also sent to another CDRM when a session is established by the SNA ACTCDRM command. The PU or cross-domain resource manager can thereby identify the SSCP with which it is in session. The identifier has the following form:

Bits 0-7:	X'05'
Bits 8-31:	X'000000'
Bits 32-47:	SSCPID in binary

The Buffer Pool Start Options

Meaning of the Options

pool name=(baseno,bufsize,slowpt,F,xpanno,xpanpt)
describes a buffer pool used by VTAM for holding data or building control blocks. This set of options defines fixed-length buffer pools. MVS VM The "F" option applies to MVS and VM only.

pool name

is the name of the fixed–length buffer pool to which these options apply. Refer to Figure 33 on page 225 for a list of the buffer pool names and, for each one, a description of its function and whether it is normally located in fixed or pageable storage.

Pool Name	Systems	Use
APBUF	VM	Large buffer pool in virtual storage
CRPLBUF	MVS, VM	RPL–copy pool in pageable or virtual storage
IOBUF	MVS, VM	Message pool in fixed storage
LFBUF	all	Large buffer pool in fixed storage (serves as message pool)
LPBUF	all	Large buffer pool in pageable or virtual storage
SFBUF	all	Small buffer pool in fixed storage
SPBUF	all	Small buffer pool in pageable or virtual storage
WPBUF	all	Message–control buffer pool in pageable or virtual storage

Figure 33. The VTAM Buffer Pools

baseno

is the base number of buffers in the pool. The minimum number is 1; the maximum number is 32767.

Do not code a comma within the number. If you code *baseno* as 0 or more than 32767, VTAM issues an error message and prompts the operator to reenter the values for the pool.

bufsize

is the size in bytes of each buffer in the pool. The only buffer pools for which the *bufsize* value can be changed are the fixed–storage message pools: MVS VM IOBUF; VSE LFBUF. For all other buffer pools, the *bufsize* value is ignored.

If a channel–attached NCP is in this domain, the *bufsize* value for the fixed–storage message pool must match the value used for the UNITSZ operand in the HOST statement in the NCP definition. If the values do not correspond, the NCP cannot be activated.

The minimum and maximum values for *bufsize* are different for different buffer pools. If the value of *bufsize* is less than the minimum, VTAM increases the value to the minimum value and issues a message to the VTAM operator. If you code *bufsize* as 0 or more than the maximum, VTAM issues an error message and prompts the VTAM operator to reenter the value for the buffer pool.

slowpt
provides a minimum set of buffers available only for priority requests, for example, read from a channel-attached device. When the number of unused buffers in this buffer pool is equal to or less than this number, buffers are allocated only for priority requests; normal requests are queued or are rejected with a return code. This is called slowdown mode.

For buffer pools to which no programs make priority requests, this operand has no meaning. Therefore, if *slowpt* is greater than 0, you are wasting buffers.

If you choose too high a value, you will never come out of slowdown mode to handle normal requests.

MVS VM **F**
tells VTAM that a buffer pool is to be MVS fixed in storage or VM locked in storage. The F attribute 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. If you intend to use the *xpanno* and *xpanpt* operands, but not the F attribute, then you must code the double comma between the *slowpt* operand and the *xpanno* operand. For example:

LPBUF=(*baseno,bufsize,slowpt,,xpanno,xpanpt*)

applies whether the buffer pool's default attribute is fixed or pageable.

xpanno
is the number of buffers, in decimal, that VTAM is to acquire when expanding the buffer pool. This value is rounded upward to the number of buffers that will fill the nearest whole page of storage. The value must be between 0 and 32767. If you code 0, dynamic buffering will not be performed. If you do not code *xpanno*, a value of 1 is used.

The use of an *xpanno* that is too small for a pool that has frequent expansions and contractions results in inefficient use of storage.

xpanpt
is a decimal integer that determines the expansion point for this buffer pool. When the number of free buffers in the buffer pool falls to a value that is equal to or less than *xpanpt*, VTAM schedules an asynchronous routine to expand the buffer pool by the number of buffers defined by *xpanno*. The value of *xpanpt* must be greater than the value of *slowpt*, but less than the difference between *baseno* and *adjval*, where *adjval* is an adjustment value for this buffer pool.

If you code an *xpanpt* value but omit the *slowpt* value, make sure that the *xpanpt* value is greater than the default *slowpt* value for the pool. If you code a *slowpt* value but omit the *xpanpt* value, make sure that the *slowpt* value is less than the default *xpanpt* value for the pool.

For example, suppose you want the large pageable-storage buffer pool to be in fixed storage with a base allocation of 12 buffers, and have the IBM – supplied value for the buffer pool size, a slowdown point at 2 buffers, an expansion size of 1 buffer, and an expansion point at 3 buffers.

MVS VM Code:

LPBUF = (12,,2,F,1,3)

See *Network Program Products Storage Estimates* for the default values.

VSE Code:

LPBUF = (12,,2,1,3)

because pageable buffer pools cannot be in fixed storage.

See *Network Program Products Storage Estimates* for the default values.

VSE **variable – length poolname = vbsz**
is the size in bytes of the variable – length buffer pools, where *variable – length poolname* represents VFBUF or VPBUF. The minimum size is 1 byte; the maximum size is 16777215 bytes. If the value is not a multiple of 2048 bytes, VTAM rounds it up to the next multiple of 2048.

vbsz is the only option you can code for a variable – length buffer pool. The variable – length buffer pools are:

VFBUF	variable – length buffer pool in fixed storage
VPBUF	variable – length buffer pool in pageable storage

See *Network Program Products Storage Estimates* for the default values.

Overriding Buffer Pool Values

All buffer pool values that you code override the previously entered values, while values you do not code default to the IBM – supplied values.

In the following examples, you have defined overriding LPBUF values in the default start option list. When VTAM is started, the system operator selects start option list ST. This list contains new LPBUF values. VTAM processes these values in the order shown below. For more details on processing, see "The LIST Start Option" on page 233.

MVS ***Example:*** In this example, the operator enters LIST = ST.

```
IBM default              LPBUF=(64,1016,0,,1,1)
ATCSTR00                 LPBUF=(128,,1,,0,0)
ATCSTRST (LIST=ST)       LPBUF=(8,,0,F,,)
```

The result is:

```
LPBUF=(8,1016,0,F,8,1)
```

VSE ***Example:*** In this example, the operator enters LIST=ST. (The F option does not apply in VSE.)

```
IBM default              LPBUF=(15,1016,0,1,1)
B.ATCSTR00               LPBUF=(128,,1,0,0)
B.ATCSTRST (LIST=ST)     LPBUF=(8,,0,8)
```

The result is:

```
LPBUF=(8,1016,0,8,1)
```

VM ***Example:*** In this example, the operator enters LIST=ST.

```
IBM default              LPBUF=(15,1016,0,,1,1)
ATCSTR00                 LPBUF=(128,,1,,0,0)
ATCSTRST (LIST=ST)       LPBUF=(8,,0,F,8)
```

The result is:

```
LPBUF=(8,1016,0,F,8,1)
```

Note: The double commas in the above examples denote values that are not coded.

The CDRSCTI Start Option

CDRSCTI=n|480

is the minimum amount of time, in seconds, that the system will retain dynamically-defined cross-domain resources after the last session with a cross-domain resource is terminated. *n* must be a decimal integer between 0 and 32767.

If you omit CDRSCTI from the START command and START list, a 480-second default is assumed.

The COLD|WARM Start Option

COLD|WARM

is the status to which the configuration restart facility of VTAM is to restore each major node in the predefined configuration list referred to by the CONFIG start option.

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 issue the equivalent of VARY NET,ACT,ID=*major node name*,WARM for each major node identified by the CONFIG start option.

Refer to the description of the activation of specific major nodes in *VTAM Operation* for information on the effect of the COLD and WARM options.

Note: VM COLD and WARM do not apply in V3R1.

The CONFIG Start Option

CONFIG=xx|<u>00</u>|name

names a list of major nodes to be activated when VTAM is started. You can code this option within a predefined list of start options named by the LIST start option, or the system operator can select it when starting VTAM.

xx

is any 1 or 2 alphanumeric characters that identify the file (or member) in the VTAM definition library that contains a list of major nodes to be activated when VTAM is started. You can file the list of major nodes under the name MVS VM ATCCONxx; VSE B.ATCCONxx. More than one list of major nodes can be filed to give the user a choice of configurations and to avoid having to issue a separate VARY command to activate each major node.

00

is the default value for the CONFIG start option. A user defined configuration list of major nodes can be filed under the name MVS VM ATCCON00; VSE B.ATCCON00. VTAM always uses this predefined list unless the system operator uses the CONFIG option to select another list or the CONFIG option is included in the set of LIST start options. 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=xx replaces the default start option list. This differs from start option list processing where LIST=xx, if coded, is merged with the 00 list.

MVS VSE name

is the 3- to 8-character file name of the configuration restart VSAM file containing a list of major nodes that were active and the DRDS files and sets of PATH statements that were applied at the time of failure or deactivation.

VTAM must have used the file previously to record a list of active major nodes; that is, the file name must have been coded in the NODELST option during some previous execution of VTAM.

If the file is empty, VTAM does not activate any major nodes during startup. The VTAM operator is prompted for a password if one is required to gain access to the file.

- 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.

MVS VM The CSALIMIT Start Option

CSALIMIT = 0|n|nK|nM

is the maximum amount of common service area (CSA) that can be used by VTAM. This number can be expressed as a number followed by a K, or a number followed by an M. The largest possible CSALIMIT is MVS/XA 2 billion bytes (expressed as 2048M); MVS/370 VM 16 megabytes (expressed as 16M).

0

is the default value. If you code 0 or take the default, no limit is enforced on the amount of CSA used by VTAM.

n|nK

n is the number of 1–kilobyte increments, in decimal, 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.

nM

n is the number of 1–megabyte increments, in decimal, that is the maximum amount of CSA that can be used by VTAM.

If neither M nor K is coded, K is assumed.

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, you must define LPBUF so that it does not have to expand. Other possible consequences are lost messages or failures when sessions are being initiated or terminated.*

MVS/XA The CSA24 Start Option

CSA24 = 0|n|nK|nM

is the maximum amount of 24–bit addressable common service area (CSA) that can be used by VTAM. This number can be expressed as a number followed by a K, or a number followed by an M. The largest possible value for CSA24 is 16 megabytes (expressed as 16M).

0

is the default value. If you code 0 or take the default, no limit is enforced on the amount of 24-bit addressable CSA used by VTAM.

n|nK

n is a the number of 1-kilobyte increments, in decimal, 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.

nM

n is the number of 1-megabyte increments, in decimal, that is the maximum amount of CSA that can be used by VTAM.

If neither M nor K is coded, K is assumed.

MVS VM The DLRTCB Start Option

DLRTCB=n|32

is the largest number of task control blocks (TCBs) to be used by dump-load-restart subtasks. *n* is a decimal integer from 1 through 64. If you code 0, VTAM uses the default value of 32. If you code a value larger than 64, VTAM issues an error message and prompts the VTAM operator to re-enter the DLRTCB value.

DLRTCB=n|8 VM V3R1

is the largest number of task control blocks (TCBs) to be used by dump-load-restart subtasks. *n* is a decimal integer from 1 through 64. If you code 0, VTAM uses the default value of 8. If you code a value larger than 64, VTAM issues an error message and prompts the VTAM operator to re-enter the DLRTCB value.

The HOSTPU Start Option

HOSTPU=host subarea PU name|ISTPUS

is the network name of VTAM host subarea PU contained in this host. The VTAM host subarea PU name allows you to uniquely identify the host PU containing channel-attached, or application resources. It is recommended that you code HOSTPU if you are using NetView or NLDM. If you do not code HOSTPU, VTAM uses ISTPUS as the host PU name.

Note: 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. This host PU name should be network-unique.

VM HOSTPU does not apply in V3R1.

The LIST Start Option

The HOSTSA Start Option

HOSTSA=n|1

is a decimal integer chosen by the user to identify this host VTAM's subarea. This start option must be coded for any VTAM that engages in cross-domain communication. *n* is an integer from 1 through 255. When this host VTAM is active, and a part of a multiple-domain network, no other subarea in the multiple-domain network that has the same subarea number can also be active.

The IOINT Start Option

IOINT=n|180

is the time interval (in seconds) after which any outstanding response for most request units sent by the SSCP, physical unit services, or logical unit services of VTAM, will be identified with a VTAM message. It must be a decimal integer from 0 to 5366000. Coding IOINT=0 deactivates the function. If you do not code IOINT, the default of 180 is assumed. If you code a value less than 60, IOINT=60 (1 minute) is assumed.

The ITLIM Start Option

ITLIM=n|0

is the maximum number of session services requests, USS requests, and VTAM macro requests that VTAM can process simultaneously. (Examples of VTAM macro requests are SIMLOGON and CLSDST. VM In V3R1, VTAM macro requests are not counted against the ITLIM total.)

These maximums apply to cross-network sessions, cross-domain sessions, and same-domain sessions. *n* is a decimal integer from 0 to 65535.

If VTAM receives more requests than allowed by ITLIM, these requests are queued within VTAM and are processed after the current requests have completed.

By placing a limit on the number of session services, USS, and macro requests that VTAM can process concurrently, the storage allocations needed to process these requests can be distributed over a longer period of time. This distribution of storage allocations will reduce the total amount of storage needed for domain start-up and shutdown.

If you code 0 or take the default, VTAM will not restrict the number of concurrent requests and will schedule each request for processing as soon as the request is received. See *Network Program Products Planning* for recommendations on defining ITLIM and storage.

The LIST Start Option

LIST=yy

determines which list of predefined start options is used to initialize a specific VTAM domain. Each ATCSTRyy list is filed in the VTAM definition library and the user can choose any two alphanumeric characters (for *yy*) to identify user-defined lists of start options.

LIST=yy can be entered only by the VTAM operator when starting VTAM. The default start list, ATCSTR00, cannot be entered by the VTAM operator and is always processed. If the VTAM operator enters more than one LIST start option, VTAM processes only the last LIST start option that was entered.

VM The MAXAPPL Start Option (V3R1 Only)

MAXAPPL=n|10

is the largest number of application programs (APPL statements) that may be active concurrently. This start option applies only to VTAM V3R1 running on a VM system. VTAM uses this value to allocate fixed storage for control of application programs. *n* is decimal integer from 1 up to the maximum MAXAPPL value. The maximum MAXAPPL value is related to the MAXSUBA value, as shown by the following table:

MAXSUBA Value	Maximum MAXAPPL Value
3	16378
4 through 7	8186
8 through 15	4090
16 through 31	2042
32 through 63	1018
64 through 127	506
128 through 255	250

If the value of MAXAPPL exceeds the maximum value allowed, VTAM issues an error message and prompts the VTAM operator to correct the value. If the value of MAXAPPL is too small in proportion to the number of ACBs to be opened, the opening of some ACBs is unsuccessful because not enough storage is available.

If the value of MAXAPPL is too large in proportion to the number of ACBs to be opened, an excessive amount of virtual storage is reserved for controlling application programs, and the ability to configure channel-attached devices and establish parallel sessions is diminished.

If you do not code MAXAPPL, the default value is 10.

Note: For each application program defined as being capable of parallel sessions (PARSESS=YES in the APPL statement), increase the value of MAXAPPL by one.

The MAXSUBA Start Option

MAXSUBA for Communicating with V3R1 VM and Pre–Version 3 Nodes

MAXSUBA=n|15 (for communicating with V3R1 VM and pre–Version 3 nodes)

n is the highest SUBAREA value, in decimal, that can be assigned to any pre–Version 3 node within this network. A pre–Version 3 node is either a host processor running a release of VTAM, *or* a communication controller running a version of NCP, for which extended network addressing is not supported. For VTAM this includes V3R1 for VM and all releases prior to Version 3. For NCP this includes all releases prior to Version 4.

n is an integer between 3 and 255. If you code MAXSUBA, the range of its values must be the same for all nodes in the same network. That is, the MAXSUBA values must all use the same number of bits.

In networks where there are no pre–Version 3 nodes, or where you do not want to communicate with any such nodes, you do not need to code MAXSUBA.

See *Network Program Products Planning* for considerations when using MAXSUBA.

MAXSUBA For V3R1 VM and Pre–Version 3 Nodes

MAXSUBA=n|15 (for V3R1 VM and pre–Version 3 nodes)

n is the highest SUBAREA value, in decimal, that can be assigned to any V3R1 VM node or pre–Version 3 node in this network. A V3R1 VM node is a host processor running VTAM V3R1 under VM. A pre–Version 3 node is either a host processor running a version of VTAM, *or* a communication controller running a version of NCP, for which extended network addressing is not supported. For VTAM this includes V3R1 for VM and all releases prior to Version 3. For NCP this includes all releases prior to Version 4. *n* is an integer ranging from 3 through 255.

If you code MAXSUBA, the range of its values must be the same for all nodes in the same network. That is, the MAXSUBA values must all use the same number of bits.

VM This start option is required for V3R1.

The MSGMOD Start Option

MSGMOD=YES|NO

YES

tells VTAM to insert the last 5 characters of the name of the VTAM module that issued a VTAM message into each VTAM message written. Note that messages may be truncated.

NO

tells VTAM not to identify the issuing VTAM module in a VTAM message. If you do not code MSGMOD, MSGMOD = NO is assumed.

MVS VM The NETID Start Option (Interconnection only)

NETID = network id

is the 1– to 8–character name of the network containing the host. NETID should be unique within a set of interconnected networks.

Using NETID in a system with just one network allows you to use the name translation facility. See "Alternatives to Predefining Cross–Domain Destination Logical Units" on page 215.

In an interconnected network, the network ID allows VTAM to determine which gateway NCP definition statements apply to the gateway host. It is also used by the gateway SSCP when sending and receiving cross–network requests.

Both NETID and SSCPNAME are required in a gateway configuration. If you code one without the other you will receive a reminder at initialization that you must code the missing operand if VTAM is to serve as a gateway SSCP.

For more detailed information, see "The NETID Start Option" on page 296.

The NODELST Start Option

NODELST = name

is the name of a file in which VTAM is to maintain a list of all currently active major nodes.

After the halting of VTAM or the failure of the host operating system, the host processor, or VTAM, the VTAM operator can reactivate the major nodes or the dynamic reconfiguration data set that was active at the time of deactivation or failure by coding CONFIG = name in the VTAM start procedure, where *name* is the NODELST name that was in effect when VTAM was last running.

The VTAM operator is prompted for a password if one is required.

If NODELST and CONFIG have the same value when VTAM is started, VTAM attempts to activate the major nodes listed in the configuration restart data set. This same data set is then used as the NODELST data set, and is updated when major nodes are activated or deactivated.

If you code NODELST and VTAM is unable to open the data set, the configuration list will not be processed.

If the name in NODELST is different from the name in CONFIG when VTAM is restarted, the data set identified by NODELST is erased before any major nodes are activated.

The PPOLOG Start Option (V3R1.1 Only)

PPOLOG = YES|<u>NO</u>
determines how much information is to be recorded in the primary program operator (PPO) log. When a primary program operator 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
- VTAM messages that are not related to those commands.

Note: When NetView is running, it is always the primary program operator.

The PPOLOG start option allows you to record these additional events in the PPO log:

- Commands entered at the system console
- Messages issued by VTAM in response to those commands.

YES
means that all VTAM commands entered at the system console (except START and HALT) and all messages VTAM issues in response are to be recorded in the PPO log.

NO
means that VTAM commands entered at the system console and messages VTAM issues in response are *not* to be recorded in the PPO log.

You can code PPOLOG = YES *only* if NetView is installed in your system. If you do not code PPOLOG, PPOLOG = NO is assumed.

The PROMPT Start Option

<u>PROMPT</u>|NOPROMPT
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. PROMPT means that VTAM is to prompt the VTAM operator to enter VTAM start options. 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 the VTAM operator will never want to change the VTAM start options.

When entering start options from the console due to a prompt from VTAM, if all the desired options cannot be entered on one line, place a comma after the last option on the line of input, and VTAM will prompt you for more start options.

The SONLIM Start Option

SONLIM = ([m|60][,t|30])

is the maximum number of fixed I/O buffers available for the session outage notification (SON) request units. The buffers controlled by SONLIM may contain SON request units for routes between the host processor and:

- Channel–attached cluster controllers
- Channel–attached communication controllers
- Channel–attached host processors.

VTAM calculates the *m* number of buffers and the *t* value, using the values you code 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 buffering. When a virtual route fails, VTAM determines the number of buffers that may be allocated for these request units by using the maximum value calculated when VTAM was started. VTAM informs each session using the failing virtual route that the session is no longer usable.

This process does not preallocate buffers. When no virtual route failures are being processed, no I/O buffers are allocated.

m|60

is the percentage of the fixed I/O buffer pool that may be allocated for session outage notification request units. It does this by using this *m* percentage and a value coded for IOBUF. When a virtual route fails, this is the maximum number of buffers that may be allocated for notification request units. The value must be a decimal integer from 1 to 99. The default is 60. If VTAM cannot initiate all the SON request units (one buffer is required for each request unit) within the available storage, it will suspend 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.

t|30

is the percentage of fixed I/O buffers that are still allocated to session outage notification request units when the notification process begins again. Once the number of buffers allocated to session outage notification drops below this *t* value, the process begins generating SON reports again. The value must be a decimal integer from 1 to 99. The default *t* value is 30.

Note: The value of *t* must be less than the value of *m*.

The SSCPDYN Start Option

SSCPDYN = YES|NO

determines whether VTAM is to add entries dynamically to the adjacent SSCP table.

If SSCPDYN = YES, the default, 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.

If SSCPDYN = NO, VTAM does not add new entries *unless* it knows the owner of the resource. This is determined by the CDRM operand of the CDRSC definition statement, or by session initiation.

This start option is closely related to SSCPORD. "Using SSCPDYN and SSCPORD to Improve Performance" on page 281 tells how you can use them together to get optimum performance.

VM SSCPDYN does not apply in V3R1.

For more details on SSCPDYN itself, see "The SSCPDYN Start Option" on page 296.

MVS VM The SSCPNAME Start Option (Interconnection only)

SSCPNAME = name

is the name of the VTAM SSCP. It is required only if VTAM is to be a gateway SSCP; although it is recommended for all SSCPs.

Both NETID and SSCPNAME are required in a gateway configuration. If you code one without the other you will receive a reminder at initialization that you must code the missing operand if VTAM is to serve as a gateway SSCP.

See "The SSCPNAME Start Option" on page 297 for SNA network interconnection considerations concerning this start option.

The SSCPORD Start Option

SSCPORD = PRIORITY|DEFINED

determines whether VTAM is to search an adjacent SSCP table in priority order (the default) or in the order in which the table is defined.

Priority order means that VTAM gives preference to SSCPs for which the most recent session initiation attempt succeeded, or for which no sessions have been tried.

Defined order means that VTAM scans the table in the same order that SSCPs are listed, regardless of whether past session initiation attempts were successful.

This start option is closely related to SSCPDYN. "Using SSCPDYN and SSCPORD to Improve Performance" on page 281 tells how you can use them together to get optimum performance.

VM SSCPORD does not apply in V3R1.

For more details on SSCPORD itself, see "The SSCPORD Start Option" on page 298.

The SUPP Start Option

SUPP = <u>NOSUP</u>|INFO|WARN|NORM|SER

is 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. After VTAM has been started, the VTAM operator can use the SUPP operand of the MODIFY command to suppress messages.

If multiple console support (MCS) has been included in the system, all suppressed messages are sent to the hard-copy log.

The SUPP start option and the SUPP operand of the MODIFY command are identical in form and effect.

NOSUP

means that all VTAM messages are printed at the console (NOSUP means "no suppression").

INFO

means that informational messages are suppressed. Informational messages are those that tell the operator that commands or procedures have been accepted for processing.

WARN

means 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 tell the operator that there is a problem, such as an invalid operand or a minor node that cannot be activated, but that VTAM can continue to process other parts of the command or procedure.

NORM

means that normal completion messages (as well as informational and warning messages) are suppressed. Normal completion messages tell the operator that commands have completed processing successfully, that a configuration has been activated successfully, that a procedure has been terminated, and so on.

SER

means 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. Messages that are generated in response to an operator request (such as the DISPLAY command) and messages that require a response (prompting messages) also cannot be suppressed.

The TNSTAT Start Option

TNSTAT[,CNSL|NOCNSL][,TIME=n|60]|NOTNSTAT

TNSTAT
means that tuning statistics should be kept.

NOTNSTAT
means that tuning statistics should not be kept.

If you code neither TNSTAT nor NOTNSTAT, NOTNSTAT is assumed.

Note: If you do not code TNSTAT at start time, you cannot use the MODIFY command to request that tuning statistics be kept at a later time.

The following operands are valid only if you code TNSTAT:

CNSL
means that the tuning statistics records are written to the console as well as to the MVS System Management Facilities (SMF) data set; VSE trace file; VM VTAM TUNSTATS file. For more information see *VTAM Customization.*

NOCNSL
means that the tuning statistics records are written only to the MVS SMF data set; VSE trace file; VM VTAM TUNSTATS file.

TIME=n
is the number of minutes that should elapse between records. One record is written every *n* 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 or MODIFY NOTNSTAT command is issued.

The value for TIME must be from 1 to 1440. If you do not code TIME, TIME=60 is assumed.

Notes:

1. *If you do not code CNSL or NOCNSL, NOCNSL is the default.*

2. 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.*

The Buffer, I/O, NCP Line, or SMS TRACE Start Option

TRACE|<u>NOTRACE</u>,ID=nodename,TYPE=BUF|IO[,EVERY]
TRACE|<u>NOTRACE</u>,ID=linename,TYPE=LINE[,COUNT=n|<u>ALL</u>]
TRACE|<u>NOTRACE</u>,ID=nodename,TYPE=SIT[,COUNT=n|<u>ALL</u>]
TRACE|<u>NOTRACE</u>,ID=VTAMBUF,TYPE=SMS

TRACE
tells VTAM to start a specific type of trace for a specific node or for monitoring the usage of all the VTAM buffer pools. Once started, the trace remains in effect until it is stopped either by stopping VTAM or by the VTAM operator entering the MODIFY command with the NOTRACE operand. More than one trace can be in effect at the same time; however, you must code a separate TRACE start option to start each trace.

NOTRACE
tells VTAM to cancel the trace requested in the ID and TYPE operands. You should code this start option only when it is necessary to override a predefined TRACE start option. You must code a separate NOTRACE start option to stop each trace.

ID=nodename
names the specific node for which a VTAM trace is to be started or stopped. *nodename* corresponds to the name assigned to the node and filed in the VTAM definition library. See *VTAM Diagnosis Guide* for a complete list of the types of nodes you can choose. 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 to trace all PUNS activity (PU to PU).

Each terminal or LU to be traced must be explicitly coded in a TRACE start option.

TYPE=BUF|IO
is the specific type of VTAM trace that is to be started or stopped for a node. You can request only one type of VTAM trace in each TRACE start option (whether entered by the VTAM operator or predefined in a list of start options). In the TYPE operand, BUF is coded for a VTAM buffer trace, or IO for a VTAM I/O trace.

EVERY
tells VTAM that the requested trace is to be started or stopped for the resource named in the ID=operand and all appropriate minor nodes of this resource (that is, all minor nodes for which I/O is done). You may

also abbreviate this operand as E. This operand pertains only to buffer and I/O traces. For example:

TRACE,ID = linename,TYPE = IO,E

This start option will initiate 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, you must code the EVERY option. You can name the link in a channel – attachment major node can be named, but not the link station.

ID = linename
names the specific line for which a VTAM trace is to be started or stopped. *linename* corresponds to the name on the LINE statement that represents the same line.

Each line to be traced must be explicitly coded in a TRACE start option.

TYPE = LINE
tells VTAM that an NCP line trace is to be started or stopped for a communication controller. There can be up to eight concurrent line traces and scanner interface traces for each NCP, depending on how each NCP was generated.

TYPE = SIT
tells VTAM that a scanner interface trace (SIT) is to be started for a 3725 Communication Controller. There can be up to eight concurrent line traces and scanner interface traces for each NCP, depending on how each NCP was generated.

COUNT = n|ALL
is the number of bytes of data you want traced. *n* is a decimal number between 0 and 254. If you do not code COUNT, VTAM traces the entire path information unit (PIU).

COUNT is valid only if you are running a scanner interface trace (SIT) or LINE trace for a 3725 Communication Controller.

ID = VTAMBUF,TYPE = SMS
tells VTAM to monitor 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 were being used at that time.) If you code TYPE = SMS, you must also code ID = VTAMBUF.

The VTAM Internal TRACE Start Option

TRACE,TYPE=VTAM
[,MODE=INT|EXT]
[,OPTION=ALL|option|(option[,option...])]
[,SIZE=n|2]
NOTRACE,TYPE=VTAM

NOTRACE is the default. (VM In V3R1, TRACE is the default.)

TRACE,TYPE=VTAM
requests a VTAM internal trace when VTAM is started. TYPE=VTAM must be the first operand after the word TRACE; otherwise the start option is ignored and you get the default, NOTRACE.

You do not have to code TRACE,TYPE=VTAM in a start option list (ATCSTRxx) to be able to use the VTAM internal trace. You can activate it using the MODIFY command, even if NOTRACE is coded in the start option list.

Note that having the internal trace buffer active uses at least 4K bytes of storage. Therefore, you should only activate the trace when necessary.

NOTRACE,TYPE=VTAM
means an internal trace will not be started. Note that if this is coded, all other operands are invalid.

MODE=INT|EXT
determines that tracing is to be recorded on an external device (EXT) or on an internal trace table (INT). MODE=EXT prevents loss of information because of wraparound in the internal trace table.

MVS You can record all VTAM traces by the Generalized Trace Facility (GTF). You can format output using the PRDMP service aid. GTF must be active when traces are initiated.

VSE You can run the VTAM trace print utility (TPRINT) as a subtask under VTAM or as a job step under VSE. This program records trace records in wraparound fashion either in main storage or in a trace file assigned to a disk or tape. If the trace is on a disk or tape, the operator is notified when the end of the trace file is reached. When a trace file is full, the oldest records are overlaid by the new trace records. Therefore, if trace data is being produced faster than the trace utility can print it, trace data is lost, due to wraparound.

VM You can issue the CPTRAP command to enable the recording of GCS and VTAM virtual machine trace data in the CPTRAP spool file. The VM TRAPRED service routine will format trace data in the CPTRAP spool file using a VTAM exit routine. ACF/TAP, which is part of the NCP System Support Product (SSP), will provide extended formatting for VTAM and NCP trace records.

The VTAM Internal TRACE Start Option

See the *VTAM Diagnosis Guide* for a more detailed description of printing trace output.

OPTION=ALL|option|(option[,option...])
is a listing of the VTAM functions that you wish to trace. This operand may be abbreviated as OPT. You may code one or more of the operands listed below.

ALL	All functions listed below
API	Application program interface
CIO	Channel I/O trace
LOCK	Locking
PSS	Process scheduling services
SMS	Storage management services
PIU	Path information units
MSG	Messages
SSCP	System services control point

If you omit the OPTION operand, only error records are traced. VM In V3R1, if you omit the OPTION operand, the following functions are traced: API, PIU, and MSG.

SIZE=n|2
is the size of the MVS VM internal trace table; VSE external trace table. The table records the use of VTAM resources for the VTAM internal trace. Use this option to choose the number of pages to be allocated from the trace table. You may code decimal integers from 1 to 999. If you omit this option, the default is 2.

If you do not choose a large enough value for the SIZE option, valuable information may be destroyed because of wraparound in the trace table. If you use this operand to select a different size on a subsequent command, information will be lost because the trace table is freed so that another table with a new size can be obtained.

Note: API, PIU, SSCP, and SMS trace records are written when an error condition is encountered and any internal trace option is active, whether or not the option for those records is in use.

VSE 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 to be allocated depends on the number of 2048-byte records that the user wishes preserved. The VTAM trace facility is described in more detail in *VTAM Diagnosis Guide.*

The USSTAB Start Option

USSTAB=tablename
is the name of a USS table to be used for VTAM operator messages and for VTAM operator commands that are processed through USS. *tablename* is the 8-character name of a USS table. You can choose one of the following USS tables:

- ISTINCNO, which contains the IBM-supplied VTAM messages and commands. ISTINCNO is the default. This table may be supplemented with a user-defined table.
- 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 has been modified. Commands processed through USS will continue to use ISTINCNO.
- A user-defined table that contains user-modified messages and commands. Messages and commands not defined in the user-defined table will continue to use ISTINCNO.

VM The VTAMEAS Start Option (V3R1 Only)

VTAMEAS=n|404
is the maximum number of concurrently active SNA network addressable units and channel-attached non-SNA terminals. This start option applies only to VTAM V3R1 running on a VM system. VTAM uses this operand in a lookup scheme to find a representation of a session between the host VTAM and an SNA network addressable unit or channel-attached non-SNA terminal.

n is a decimal integer from 0 through 32767.

Choose a number that is 10 to 20 percent greater than the total number of SNA network addressable units and channel-attached non-SNA terminals that you expect to be concurrently active. One way to estimate the maximum number is to take the sum of the number of NCP major nodes and the number of LOCAL, LU, and PU statements used to define the network to VTAM. If the actual total number of active SNA network addressable units and channel-attached non-SNA terminals in the VTAM system is greater than the value of VTAMEAS, or if the actual number is greater than 8080, the VTAM path length is increased.

If you do not code VTAMEAS, 404 is the default value.

Chapter 7. MVS VM SNA Network Interconnection Considerations

This chapter contains special considerations for installations using interconnected SNA networks. You can use interconnection on MVS systems and on V3R1.1 VM systems. These considerations include:

- Defining VTAM resources
- Defining VTAM requirements for the NCP
- Defining VTAM start options.

General discussions of these topics, along with the formats of definition statements and start options, are in the preceding chapters. This chapter only discusses additional things you must consider for interconnection.

Defining VTAM Resources

Defining Application Program Major Nodes

The Alias Name Translation Facility

If your installation uses interconnected SNA networks and you want to use the alias name translation facility provided by NCCF or a similar user-written application, you must include an APPL definition statement for that application program in the application program major node. The name on the APPL statement must be ALIASAPL. Following is an example of an APPL statement for the alias name translation facility:

```
ALIASAPL APPL   AUTH=(CNM),PRTCT=NCCF2
```

where NCCF2 is a password for NCCF.

For complete coding information for the APPL definition statement, see "The APPL Definition Statement" on page 82.

CNM Application Programs

The AUTH=CNM operand on the APPL definition statement means that this application program can use the communication network management (CNM) interface. This operand must be coded for all CNM applications, including:

- The NCCF alias name translation facility
- The NLDM component of NetView
- The NLDM program product
- A user-written application program used for name translation.

For a complete list of IBM CNM application programs, see *VTAM Customization.*

Defining a Cross – Network Controlling Primary Logical Unit

Using the LOGAPPL operand, you can define a cross – network controlling primary logical unit for a local non – SNA device, a local SNA logical unit, and a switched SNA device.

The effect is the same as if a same – network CDRSC were the controlling primary logical unit, except that the resulting session is cross – network rather than same network.

For coding information, see the appropriate section in Chapter 5, "Defining the Network to VTAM" on page 69 for defining major nodes. *NCP – SSP Resource Definition Reference* may also be helpful.

Deferred Session Setup

In a single – network environment, when the SSCP – SSCP session necessary for the establishment of a controlling primary logical unit session is not immediately available, VTAM stores this information. When the necessary SSCP – SSCP session becomes active, VTAM will attempt to establish the LU – LU session.

You can establish cross – network controlling primary logical unit sessions through any number of intermediate SSCPs, provided that all required SSCP – SSCP sessions are active when the LU – LU session setup is attempted.

In the case of a cross – network session where the session setup path consists of more than one SSCP – SSCP session, the session setup proceeds as far as establishing the session control blocks at each active SSCP along the setup path. At some point along the path, an SSCP may determine that the session request (CDINIT) cannot be routed any farther because it has no active sessions with any SSCPs in its adjacent SSCP list for the specific destination logical unit. If this occurs, the CDINIT request is rejected, the session control blocks are freed, and the operator is notified of the failure. Once the necessary SSCP – SSCP sessions are active, the operator can issue a VARY LOGON command in the secondary logical unit's host to restart the procedure.

Sessions With Dial – Out Resources

A controlling primary logical unit for a switched SNA device may be a cross – network resource.

A CDINIT request is not the first request sent along the session setup path if the controlling primary logical unit is specified for a dial – out logical unit. CDINIT will be preceded by a DSRLST (Direct Search List) request unit, which establishes that the named controller can be reached and that the resource is capable of being a primary logical unit. This procedure prevents an unnecessary dial connection while this determination is being made. (CDINIT cannot be sent until dial – out has occurred, and the logical unit's network address is established.)

Cross – Network Controlling Primary LU

Deferred session setup is not supported for dial – out logical units controlled by cross – network primary logical units. In this case, failure to reach the controlling primary logical unit is reported to the operator. Once the necessary SSCP – SSCP sessions are active, the operator can issue a VARY LOGON to restart the procedure.

Defining Cross – Domain Resource Managers

For SNA network interconnection, a cross – domain resource manager (CDRM) major node consists of:

- A VBUILD statement
- Optional NETWORK definition statements
- One or more CDRM definition statements
- Optional GWPATH definition statements.

There are no SNA network interconnection considerations for the VBUILD statement. Considerations for the remaining statements are described below. Following this description are guidelines for defining cross – network resources that include examples of how the above statements are used.

Interconnection: NETWORK

The NETWORK Definition Statement

For SNA network interconnection, the NETWORK definition statement is added to the CDRM major node definition to name the network in which each CDRM resides.

Name	Definition Statement	Operands
[name]	NETWORK	[NETID=network id]

name

is a 1– to 8–character name that identifies this NETWORK statement in error messages from the VTAM system definition processor. To avoid confusion, use the same value for both *name* and the NETID operand.

NETID=network id

is an optional 1– to 8–character name that identifies the network in which the CDRM or set of CDRMs resides. This NETID pertains to the CDRM or set of CDRMs defined following the NETWORK statement and before any other NETWORK statement in the same set of definitions.

You may include more than one NETWORK definition statement in a CDRM major node definition. If you include consecutive NETWORK statements (without intervening CDRM definitions), only the last one will be used. That is, the network name of a CDRM is defined by the last NETWORK statement preceding the CDRM definition statement.

If no NETWORK definition statement exists before one or more CDRM definition statements, the CDRM(s) is assumed to reside within this same VTAM network. The network name for this network is the NETID of the VTAM in which the definition is filed.

Note: If VTAM does not have a network name because none was given at start time, only CDRMs with no associated network ID are processed by this VTAM. All CDRM statements associated with a specific NETID (other than the null or blank) are ignored.

The CDRM Definition Statement

Name	Definition Statement	Operands
cdrmname	CDRM	[CDRDYN=YES|NO] [,CDRSC=OPT|REQ] [,ELEMENT=n|1] [,ISTATUS=ACTIVE|INACTIVE] [,RECOVERY=YES|NO] [,SPAN=(spanname)] [,SUBAREA=n] [,VPACING=n|0|63]

Although no additional operands are needed on the CDRM statement for SNA network interconnection, additional considerations do apply to the following existing operands:

cdrmname

For CDRMs in another network, this 1 to 8 character name is the name by which this VTAM knows the other-network CDRM. This *cdrmname* is used in the ID operand of VTAM operator commands. Therefore, it must be a unique name within this VTAM. As a result, the name fields of all CDRM definition statements that are active at one time must be unique and must not conflict with other resource names in the same network. Because the CDRM statement in an adjacent SSCP table requires the SSCPNAME of an external CDRM, you may have to change the labels on the CDRM definition statements.

To make things simple for the network operator, it is recommended that you use the same *cdrmname* for the host CDRM as for the SSCPNAME start option.

CDRDYN=YES|NO and CDRSC=OPT|REQ

These operands control an SSCP's ability to dynamically define cross-domain resources. Dynamic definition has been extended to include cross-network resources.

Some situations require dynamic definition. When you define a gateway SSCP, you must code CDRDYN=YES. Item 3 on page 274 describes another instance in which you should code CDRDYN=YES.

ELEMENT=n|1 and SUBAREA=n

The alias address representing a cross-network SSCP in VTAM's network may not be available until a session with the SSCP is established. Therefore, coding these operands on a CDRM statement for an other-network SSCP is optional. If you do not code these operands, VTAM determines their values when a cross-network SSCP-SSCP session is activated.

When the gateway SSCP does not have a session with the gateway NCP used to support a cross-network session with the SSCP defined by the CDRM statement, or when the SSCP is a back-level VTAM, the SUBAREA and ELEMENT operands define the alias address of the external SSCP in the VTAM network. This address is within the gateway NCP used for the session path. The SUBAREA number may range up to 255, and the element address may range up to 32767.

The SUBAREA operand is required for CDRMs in the same network as the host that processes the definition. For cross−network sessions, it is optional.

Notes:

1. *If no GWPATH definition statements follow the other−network CDRM statement, you must code the SUBAREA operand on that CDRM statement.*

2. *If a non−gateway SSCP in this network will participate in an SSCP−SSCP session with a gateway SSCP in another network, you must code the GWNAU definition statement in the NCP generation deck to predefine an element address for the gateway SSCP. You must also code this element address on the CDRM statement defining the gateway SSCP to the SSCP in the non−native network. (See NCP−SSP Resource Definition Reference for more information on the GWNAU definition statement.)*

3. *In a network where a gateway NCP is connected to two hosts, one with a release of VTAM that supports extended network addressing (ENA) and one without, you must observe these rules if you want to use ENA:*

   - *The gateway NCP must be controlled by the host that has ENA. Code GWCTL=ONLY on the PCCU statement for that host.*

   - *Activate the CDRM session only from the host that has ENA, either by the VARY ACT command or by using a start option list.*

   - *The gateway NCP must be a release that supports ENA.*

The ELEMENT operand has a default value of 1 for both same− and cross−network CDRMs.

For coding information, see "The CDRM Definition Statement" on page 204.

The GWPATH Definition Statement

Each GWPATH definition statement defines a possible cross-network session path between the gateway host CDRM and a CDRM in another network. One or more gateway path (GWPATH) definition statements may follow a CDRM definition statement for a CDRM in another network. The order of the GWPATH statements is the order used by VTAM when selecting the path for a given session.

For an SSCP-SSCP session, each successive gateway path between two SSCPs is tried until:

1. The SSCP-SSCP session is established, or
2. All the possible session paths have been tried unsuccessfully.

For an LU-LU session, each gateway NCP (defined by a GWPATH statement) between the two logical units is tried until:

1. The requesting logical unit receives a positive response to the Initiate request for the session, meaning that the requested session partner has been located, or

2. All the possible gateway NCPs (defined by the GWPATH statements for the CDRM in another network) have been tried unsuccessfully.

Selection of Alternate Gateway Paths for LU-LU Sessions: A gateway SSCP selects the gateway NCP for a cross-network LU-LU session independently of the gateway NCP used for the cross-network SSCP-SSCP session. The gateway NCP used for the SSCP-SSCP session is always chosen from the gateway NCP list supplied by the GWPATH statements.

During session establishment of a cross-network LU-LU session, the gateway SSCP that is controlling gateway path selection gets the list of alternate gateway NCPs defined by the GWPATH statements. From this list, the gateway SSCP creates a list of GWPATH statements only for NCPs with which the gateway SSCP has a session. If the BIND RU that flows between the requesting logical unit and the session partner fails because a virtual route required by the session cannot be activated, no further gateway path selection is attempted; the requesting logical unit is informed in an appropriate exit routine. (See *VTAM Programming* for a description of LU-LU session establishment.)

If you have supplied a session management exit routine to shorten or reorder the list of gateway NCPs, the gateway SSCP uses that altered list to establish the LU-LU session path. For details, see *VTAM Customization.*

The GWPATH definition statement is required for all gateway SSCPs that initiate cross-network SSCP-SSCP sessions. If no GWPATH statements follow a CDRM statement for a CDRM in another network, you must code the SUBAREA operand on that CDRM statement. VTAM determines the gateway NCP to use from the SUBAREA value for the CDRM.

The operands on the GWPATH statement are listed here in alphabetical order for easy reference. However, you may find it easier to study and code them in this order: GWN, SUBAREA, ELEMENT, ADJNET, ADJNETSA, and ADJNETEL.

Name	Definition Statement	Operands
[name]	GWPATH	[ADJNET=adjacent network ID] [,ADJNETEL=adjacent network element|1] [,ADJNETSA=adjacent network subarea] [,ELEMENT=element address|1] [,GWN=gateway NCP name] [,SUBAREA=subarea address]

ADJNET=adjacent network id

is the ID of the network that is adjacent to the gateway SSCP's network for this cross-network session path to the CDRM through the associated gateway NCP. The gateway NCP is determined by the GWN operand or from the SUBAREA operand.

If it is not coded, the adjacent network defaults to the ID of the network defined on the preceding NETWORK definition statement.

ADJNETEL=adjacent network element|1

is the element address of the CDRM for the adjacent network. ADJNETEL may range from 1 to 32767.

You must code ADJNETEL if the actual network of the CDRM is not the same as ADJNET and the default value of 1 is not correct.

Code ADJNETEL for all GWPATH statements. Its value should be the same as that defined on the ELEMENT operand of the GWNAU definition statement.

ADJNETSA=adjacent network subarea

is the subarea address of the CDRM for the adjacent network. ADJNETSA may range from 1 to 255. Code ADJNETSA for all GWPATH definition statements.

ADJNETSA is required when:

1. The actual network of the CDRM is not the same as the adjacent network (that is, a back-to-back configuration).

   In this case, the subarea of the gateway NCP in the adjacent network is used as the gateway to the network that contains the SSCP being defined. If you considered the gateway NCP named by the GWN operand as the first one on the path to the SSCP being defined, this ADJNETSA value identifies a second gateway NCP on the session path. This second gateway NCP must contain a GWNAU definition statement for the SSCP being defined. (See "The GWNAU Definition Statement" on page 295.)

   For ADJNETSA, code the same subarea address you coded on the SUBAREA operand for the BUILD or NETWORK definition statement in the second gateway NCP. The NETID value matches the adjacent network ID.

2. The CDRM being defined is within the adjacent network *and* this host wants to be able to initiate SSCP – SSCP sessions.

   In this case, code ADJNETSA as the real subarea address of the CDRM within its network.

   If you do not code the ADJNETSA operand, it must be coded in the external CDRM's definition of the VTAM host CDRM, and the external CDRM must initiate the SSCP – SSCP session. If neither CDRM has an ADJNETSA operand, the session cannot be established.

Note: The ADJNETSA value is *not* checked against the MAXSUBA value for VTAM's network. You must ensure that the address agrees with the MAXSUBA restrictions, if any, for the adjacent network. If it does not agree, session setup requests will fail.

ELEMENT = element address|1
is the element portion 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 activates the CDRM for the other network. ELEMENT may range from 1 to 32767.

Note: Code ELEMENT and SUBAREA if this is a non – gateway SSCP that could initiate an SSCP – SSCP session with a gateway SSCP in another network. The ELEMENT address must match the ELEMENT address predefined on the GWNAU definition statement for this CDRM. (See NCP – SSP Resource Definition Reference for more information on the GWNAU definition statement.)

GWN = gateway NCP name
is the 1 – to 8 – character name of the gateway NCP that may be used to access the CDRM. It is the same as the name used to file the gateway NCP definition statements.

If you do not code this operand, 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 this case, VTAM will use the subarea number when it accesses the CDRM, without regard to the node name.

SUBAREA = subarea address
is the subarea portion of the alias address for the CDRM in the network of the gateway SSCP. The alias address is an address in the associated gateway NCP. SUBAREA may range from 1 to 255.

For a gateway SSCP, you must code SUBAREA if you do not code GWN. You may wish to 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 GWN above.)

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 activates the other – network CDRM.

Note: Code ELEMENT and SUBAREA if this is a gateway – capable SSCP that will initiate an SSCP – SSCP session with a gateway SSCP in another network.

Defining Cross – Network Resources

Single Network

In a single – network configuration, a session can be established between an application (APPL1) in HOST1 and a terminal (TERM1) owned by HOST2 as shown in Figure 34.

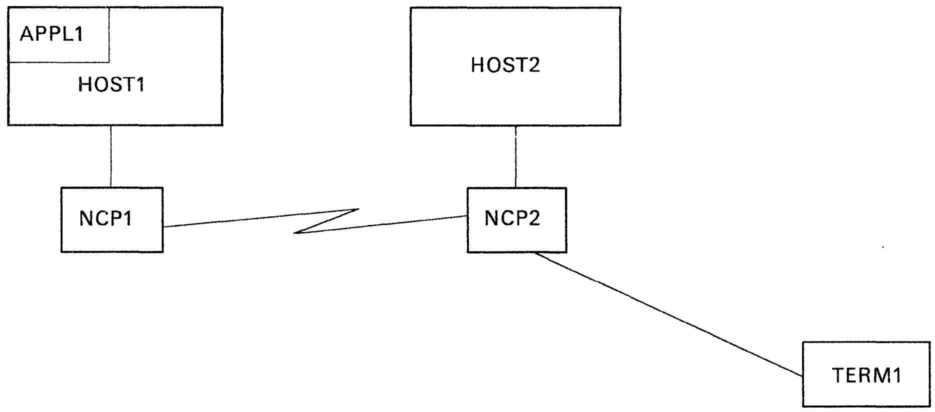


Figure 34. Example of a Single – Network Configuration

The cross – domain resource definitions in HOST1 are:

```
CDRMSEG1  VBUILD    TYPE=CDRM,...
HOST2     CDRM      ...
HOST1     CDRM      CDRDYN=YES,...

CDRSCSEG  VBUILD    TYPE=CDRSC,...
TERM1     CDRSC     CDRM=HOST2,...
```

In this example, the external CDRM (HOST2) is defined to VTAM in HOST1 using the CDRM definition. TERM1 is defined to VTAM in HOST1 using the CDRSC definition.

Similarly, in HOST2, HOST1 is defined to HOST2 using the CDRM definition. If APPL1 initiates the session with TERM1, the CDRSC definition of APPL1 to HOST2 is optional. (If TERM1 initiates the session, this definition is required, and the CDRSC definition of TERM1 in HOST1 is optional—assuming that CDRSC = OPT is coded on the HOST2 CDRM statement.)

CDRSC definition of the requested session partner—also known as the destination logical unit (DLU)—in both HOST1 and HOST2 is optional if default routing is defined and CDRSC = OPT is coded on the CDRM definition statements. See "Alternatives to Predefining Cross – Domain Destination Logical Units" on page 215. See *SNA: Concepts and Products* for details on request unit flow during session initiation.

Cross – Network Resources

Interconnected Networks

With interconnected networks, defining CDRSCs becomes more complex as cross – network sessions, alias name translation, and adjacent SSCP rerouting are introduced. The configuration in Figure 35 shows two interconnected networks.

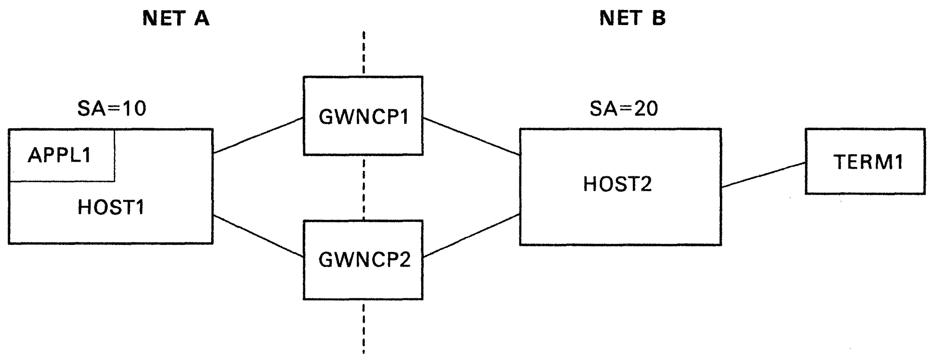


Figure 35. Example of Two Interconnected Networks

To establish cross – network sessions between APPL1 in NETA and TERM1 in NETB, the following CDRM definitions are required if sessions are to be initiated from either HOST1 or HOST2.

This CDRM major node is filed in HOST2:

```
         NETWORK    NETID=NETA
HOST1    CDRM       CDRDYN=YES,CDRSC=OPT
         GWPATH     GWN=GWNCP1,ADJNETSA=10,ADJNETEL=1,...
         GWPATH     GWN=GWNCP2,ADJNETSA=10,ADJNETEL=1,...
```

This CDRM major node is filed in HOST1:

```
         NETWORK    NETID=NETB
HOST2    CDRM       CDRDYN=YES,CDRSC=OPT
         GWPATH     GWN=GWNCP1,ADJNETSA=20,ADJNETEL=1,...
         GWPATH     GWN=GWNCP2,ADJNETSA=20,ADJNETEL=1,...
```

In an interconnected network configuration, there are three ways of defining CDRSCs:

- Method 1: dynamic allocation of CDRSCs

  Instead of predefining CDRSCs, you can decide whether some or all CDRSCs will be dynamically – allocated and defined. The host that owns the logical unit initiating a session (the origin logical unit or OLU) can dynamically define the requested session partner—the destination logical unit.

- Method 2: CDRSC definitions without the NETWORK definition statement, or CDRSC definitions with this host's NETID coded.

  This method of defining CDRSCs can be used by hosts within either of the networks in which the session partners reside. Information from the definition will be used only within the network of the session partner in which the session request originates. For example, in HOST1 (in Figure 35), you could code:

```
CDRSCSEG   VBUILD      TYPE=CDRSC,...
TERM1      CDRSC       [CDRM=HOST2,]...
```

Note: You cannot code brackets. The brackets are used here to show that, in this example, coding of the CDRM operand is optional.

- Method 3: CDRSC definitions with the owning hosts's NETIDs coded on NETWORK definition statements.

  This method of defining CDRSCs can be used by any of the hosts involved in establishing the requested session. For example, in HOST1 (in Figure 35 on page 260), you could code:

```
CDRSCSEG    VBUILD      TYPE=CDRSC,...
            NETWORK     NETID=NETB
TERM1       CDRSC       [CDRM=HOST2,]...
```

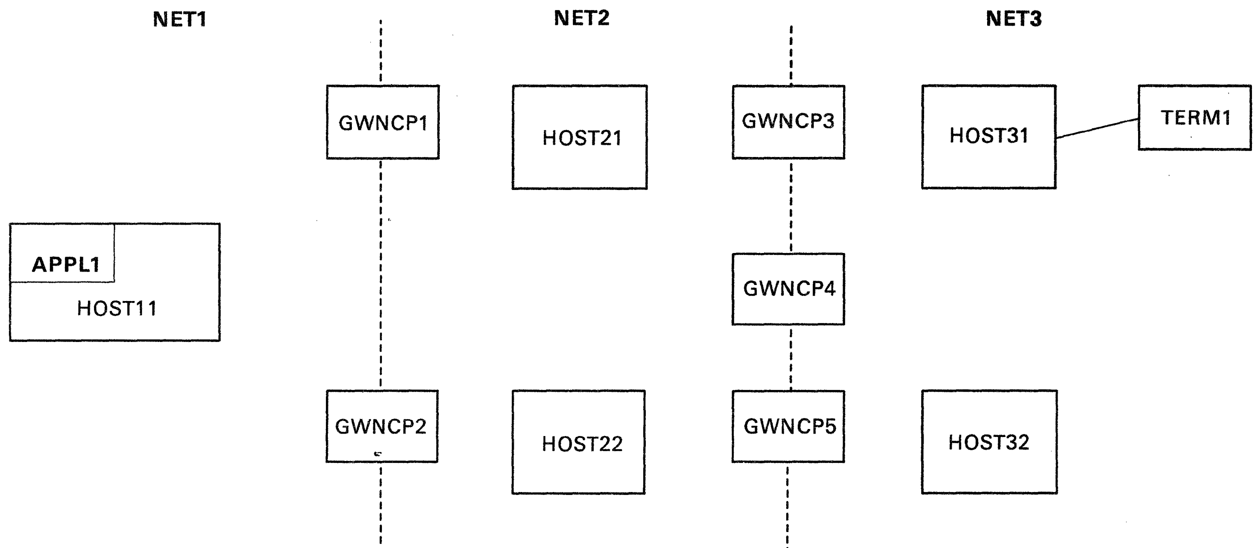
  Note: You cannot code brackets. The brackets are used here to show that, in this example, coding of the CDRM operand is optional.

VTAM offers such wide flexibility in defining (or not defining) CDRSCs because:

- The wide range of supported configurations results in different requirements for each SSCP in the configuration.
- Methods other than predefining CDRSCs can be used, such as alias name translation and default SSCP selection. These methods provide additional function as well as an alternative to predefining CDRSCs. (See the following section for a detailed explanation of alias name translation and default SSCP selection.)
- Customers are using SNA network interconnection for many reasons, for example, connecting independent networks and splitting existing networks to provide additional network addresses.

You can better understand the three methods of defining CDRSCs (and the advantages and disadvantages of each) by studying the multiple – network configuration in Figure 36 on page 262.

Cross – Network Resources



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks to the left and right of it.

Figure 36. Example of Three Interconnected Networks

If APPL1 in HOST11 attempts to establish a session with TERM1 (owned by HOST31), TERM1 (the destination LU) can be defined to HOST11 in one of the following ways mentioned earlier:

Method 1: dynamically – defined CDRSCs

Method 2: CDRSC definitions without NETID or with this host's NETID

Method 3: CDRSC definitions with the owning hosts's NETIDs

Method 1

If you choose not to predefine a CDRSC for TERM1 (and the CDRM definition for HOST11 and HOST31 is set up to allow dynamic CDRSC definition), this is what will happen:

1. A CDRSC for TERM1 is dynamically allocated in HOST11.

2. If duplicate resource names exist (for example, TERM1 exists in both NET1 and NET3), VTAM must select an alias name to refer to the session partner located in the other network. (For example, VTAM chooses an alias name by which NET1 refers to the TERM1 located in NET3.) VTAM uses the alias name translation facility to identify the network in which the session partner resides. Then, VTAM resolves the alias name to the name used in the session partner's network. Optionally, the alias name translation facility can be used to obtain the name of the SSCP that owns the session partner.

3. VTAM uses an adjacent SSCP table to locate the next SSCP to which the session setup request should be sent to reach the destination network.

Once the session – initiation request leaves the first gateway (that is, the gateway connecting NET1 and NET2), the destination logical unit's alias name, if not yet translated, is treated as the real name. Thus, VTAM assumes that the destination logical unit resides in the network of the SSCP to which the request is being rerouted.

In the configuration in Figure 36, if no translation takes place, the session initiation request means that the destination network is NET3 when the request is sent to either HOST31 or HOST32.

Advantages: Some advantages of using this method are:

- The number of definitions required in HOST11 is reduced.
- Once the real name and network are known, the lack of predefined CDRSCs does not affect the length of the session setup path.

Disadvantages: The main disadvantage of eliminating predefined CDRSCs is that the length of the session setup path is increased before VTAM identifies the destination logical unit's real name and network. In this case, a CDRSC is dynamically allocated using the alias name. Once the real name is determined—either by using the alias name translation facility or by CDINIT response—the dynamically – allocated CDRSC is replaced by a CDRSC defined with the real name and network. Note that unless an alias name translation facility that supplies the name of the SSCP owning the session partner is used, each adjacent SSCP leading to the destination network will be tried until the session is established, or it is determined that none of the SSCPs owns that requested session partner.

In the configuration in Figure 36, there are no name conflicts. If HOST11 did not have an alias name translation facility installed, the dynamic CDRSC would have been allocated by HOST11 using what it considers to be the resource's alias name: TERM1. At CDINIT response time, that CDRSC is replaced with a real representation of the CDRSC—also named TERM1.

Recommendation: Using this method of dynamically allocating CDRSCs appears to be most beneficial for intermediate and destination networks along the session setup path; the destination logical unit's real name and network identifier are known by the time the session request reaches these hosts, so replacing CDRSCs is not required. Without dynamic definition of CDRSCs, the number of resources requiring definition could be extremely large. If you provide networking services within an enterprise or for external users, this method is recommended.

Method 2

This method of defining CDRSCs involves defining CDRSCs without the NETWORK definition statement, or with the host network's NETID coded. Using the configuration in Figure 36, you could define TERM1 (the destination LU) this way:

```
          VBUILD      TYPE=CDRSC,...
TERM1     CDRSC       CDRM=HOST21,...
```

Notice that the CDRM, HOST21, is not the owning SSCP. Rather, it is the SSCP adjacent to the origin logical unit's owning SSCP (HOST11). Because the CDRSC is defined in HOST11's network (or is allowed to default), the resource name (TERM1) is assumed to be the name known in HOST11's network—that is, the alias name.

Advantages: Predefining CDRSCs using this method has some advantages:

- Coding the CDRM operand on the CDRSC definition statement allows you to select the next SSCP in the session setup path. For simple networks that do not plan to provide alternate adjacent SSCPs, this method could eliminate the need for defining adjacent SSCP tables. For more complex configurations, this method of defining CDRSCs identifies the first SSCP that should be tried when alternate adjacent SSCPs are available. In this case, VTAM reorders the adjacent SSCP table you defined; the first SSCP in the table is now the SSCP named in the CDRM operand of the CDRSC definition.

- For some configurations, new CDRSC definitions are not required. If a multiple – network configuration is the result of splitting a single network and the owning CDRMs have not changed, the CDRSC definitions previously defined still cause the request to be routed to the correct SSCP.

Disadvantages: The disadvantages of predefining CDRSCs with this method are:

- This method can only be used in the origin logical unit's network. Within the first gateway, the alias name and NETID from the origin logical unit's network are used by the gateway SSCPs to locate cross – domain resource definitions until the real name and NETID are known. Again using Figure 36, if HOST21 received a CDINIT request from HOST11 for a session between APPL1 and TERM1, HOST21 assumes TERM1 to be the resource's alias name as known in NET1. Thus, a predefined CDRSC defaulting to NET2's NETID will never be located. HOST21, then, must use dynamic CDRSC definition to define TERM1.

- The length of the session setup path is increased, although not as much as with dynamically – defined CDRSCs using the resource's alias name (since no definitions have to be built dynamically). When the real name and NETID are determined, VTAM updates the predefined CDRSC to represent the real name. When the last session for that CDRSC terminates, VTAM restores the alias name so that the CDRSC is the same as originally predefined.

Recommendation: This method of predefining CDRSCs is most suitable for simple configurations (for example, where one network was split into two) or for hosts who will rely on another host for gateway support. In our example, HOST11 passes all session requests to HOST21 or HOST22. Those hosts are then responsible for alias name translation, alias address requests, adjacent SSCP rerouting, and so forth.

Method 3

The third method of predefining CDRSCs involves including NETWORK definition statements in the CDRSC definition to identify the network in which the resource is located. For example, the following is defined in HOST11:

```
         VBUILD     TYPE=CDRSC,...
         NETWORK    NETID=NET3
TERM1    CDRSC      [CDRM=HOST31,]...
```

Note: You cannot code brackets. The brackets are used here to show that, in this example, the CDRM operand is optional.

In this example, TERM1 is defined to be in NET3. Its owning CDRM is HOST31. When a CDRSC is defined using the NETWORK statement, VTAM recognizes the owning CDRM as the actual owning SSCP—not as the next SSCP on the session setup path. A session with this CDRM is required only if it is also the next SSCP on the session setup path.

Advantages: Some advantages to using this method are:

- In a configuration with no name conflicts, this method defines the shortest session setup path. However, a resource, even when defined with a NETID, cannot be predefined if its name is not uniquely known in the host's network. This method of predefining CDRSCs was designed for those systems that want to quickly locate a resource and use its real name. Therefore, the name must be uniquely known within the network in which it is defined.

- Predefining CDRSCs in this way allows you to define very specific adjacent SSCP tables. Because the NETID and, optionally, the destination SSCP are known, you can define adjacent SSCP tables for a specific NETID *or* NETID and SSCP name. This could eliminate VTAM's need for a default SSCP list, which may not provide the shortest session setup path.

Disadvantages: Some disadvantages also exist:

- If you have coded the CDRM operand on the CDRSC definition statement, VTAM assumes it to be the owning SSCP name. Changing CDRMs in a backup situation is complex. The operator must issue a MODIFY CDRM command in each network that contains a CDRSC definition, with the previous CDRM listed as the owner. Then (using Figure 36 again), if HOST31 failed and TERM1 was acquired by HOST32, but the CDRM was not updated in HOST11, the session setup attempt would fail since HOST31 is still defined as the destination logical unit's owning CDRM. To avoid this problem of updating

CDRSCs should a backup situation arise, omit the CDRM operand; VTAM will use the default SSCP list defined for the destination logical unit's network to find the adjacent SSCP.

- This method will not work for interconnected networks with duplicate names. In many cases, renaming and predefining CDRSCs using NETID is preferable to using the alias name translation facility.

Recommendation: This method is recommended for networks that contain unique names and require fast session setup.

Initiation Request Processing

Assume that the CDRSC for TERM1 is successfully defined or allocated in HOST11. The Initiate request carries additional information when flowing between VTAM Version 2 Release 2 and later SSCPs. (Initiate requests carry a NETID if one was given when VTAM was started; otherwise, the NETID field is blank.) Assuming that HOST11 and HOST21 are Version 2 Release 2 and later SSCPs started with NETIDs given, the Initiate request carries the alias resource name and network identifier for TERM1 and, if known, the real name and network identifier. The alias name is TERM1 in NET1; the real name is TERM1 in NET3. A pair of names (real and alias) will also be carried in the request for the origin logical unit, APPL1. The origin logical unit's real name is APPL1 in NET1; the origin logical unit's alias name is APPL1 in NET3. Note that APPL1 and TERM1 are not duplicate resource names in this configuration. Therefore, the only differences between the real and alias names is the network identifier. If duplicate resource names exist, the resource name would differ, as well, for the real and alias names.

If the CDRSC for TERM1 were dynamically defined in HOST11 or predefined using its alias name, the Initiate request would carry only the alias name TERM1 in NET1. If, however, the CDRSC for TERM1 were predefined in HOST11 using its real name, the Initiate request would carry both the alias name and the real name TERM1 in NET3.

In NET2, there are two ways to define TERM1: dynamically allocate a CDRSC or predefine the resource using its real name and network identifier. As mentioned earlier, the method of predefining the resource using the alias name is used only in the originating network.

If only the alias name is passed to HOST21 in the Initiate request, a CDRSC is dynamically allocated for TERM1 in NET1 (unless one already exists from a previous session attempt.) VTAM attempts to determine the real name using the alias name translation facility, if it is available. The session setup request will then be routed to HOST31 using an adjacent SSCP table. This requires that the request leave the first gateway. Therefore, if the real name was not determined using the alias name translation facility, the real name is assumed to be the same as the alias name and the network of the adjacent SSCP to which the request is routed, that is, TERM1 in NET3. A new dynamic CDRSC is created using TERM1 in NET3, and any pertinent information is swapped from the alias CDRSC representation (TERM1 in NET1) to the real CDRSC representation (TERM1 in NET3).

Note: The request flow described above assumes that GWCTL=SHR in the host's PCCU definition statement. If, according to the rules for gateway control described later in this chapter, HOST11 were designated to request an alias address for GWN1, the real name would have been determined in HOST11.

If both the real and alias names were passed in the Initiate request, either a predefined real CDRSC representation for TERM1 in NET3 is located or a dynamic CDRSC is allocated. In either case, the resource name is not passed to the alias name translation facility, even if one is available.

In SSCP21, a resource must be defined or dynamically created for APPL1, as well. Since APPL1 is the origin logical unit, its real name and network identifier are known. A predefined CDRSC would require a NETWORK definition statement with NET1 as the network identifier. If a CDRSC were not predefined, the dynamically – created CDRSC would contain the network identifier, NET1, and the real name, APPL1.

Assuming that HOST31 is also a Version 2 Release 2 or later SSCP, the Initiate request that flows from HOST21 to HOST31 will contain both real and alias names for TERM1. HOST31 determines that it owns the destination logical unit. At that time, a CDRSC representation for APPL1 is located or is dynamically allocated using the real name (APPL1 in NET1). If APPL1 were known by an alias name in NET3, VTAM would search for the CDRSC representation for that resource using its alias name. When the representation is found, VTAM would swap any pertinent information from the alias CDRSC representation to the real representation.

If Initiation request processing is successful, HOST31 sends a positive response. When HOST11 receives the response, further processing depends on how TERM1 was defined in HOST11. If TERM1 was dynamically allocated, a new dynamic CDRSC is allocated and the information is swapped. If TERM1 was predefined using its alias name, the predefined CDRSC representation is updated with the real name, network identifier, and owning SSCP. If TERM1 was predefined using its real name and network identifier, no further processing is required. Session setup continues, and resources are recognized by their real names and network identifiers.

Which Method Should You Use?

This section contains guidelines only for predefining CDRSCs. You will have to decide which method or combination of methods is best suited to your configuration. Carefully consider the following questions:

- How many CDRSCs would you have to predefine?
- How many duplicate names exist? How many resources would you have to rename and redefine if you did not install the alias name translation facility?

Cross – Network Resources

- What is your main reason for using SNA network interconnection? Are you splitting an existing network? Or, are you joining two or more independent networks?

- Do you plan to use the alias name translation facility?

Your answers to these questions should help you decide which method of predefining CDRSCs you should use.

Alternatives to Predefining Cross-Network Destination Logical Units

There are three ways to lay the groundwork for communicating with a destination logical unit in another domain. One is to define the destination logical unit to VTAM as a cross-domain resource owned by an external CDRM, as described later in this chapter. However, you can also use either default SSCP selection or the alias name translation facility to allow communication with a cross-domain destination logical unit. These alternatives are discussed in the following sections.

Default SSCP Selection

If you have defined an adjacent SSCP table, VTAM sends the session setup request for an undefined destination logical unit to the adjacent SSCPs in the list until the owning SSCP is found or the end of the list is reached. If the adjacent SSCP to which a cross-network session setup request is routed does not own the destination logical unit, it will reroute the request, provided a list of default SSCPs exists for that SSCP and the maximum SSCP rerouting count (described in *VTAM Customization*) has not been reached. Otherwise, the request is rejected.

Once the owning SSCP is found by this trial-and-error routing of the CDINIT request, VTAM automatically creates a CDRSC for the logical unit, if dynamic definition of resources is allowed.

Performance: It is recommended that you know the NETIDs of the other-network CDRMs and put them in your adjacent SSCP table. Doing this saves time when VTAM does trial and error routing.

Using an Alias Name Translation Facility

Another alternative to predefining destination logical units to VTAM is to install a name translation facility and define to it all logical units (or, at least, those that will have cross-network sessions) in the interconnected networks.

To use this method, you must define all the logical units in the interconnected networks that are not already defined, and their owning SSCPs, to the name translation facility and assign alias names. If VTAM does not have a definition of a destination logical unit for a session setup request, it calls the name translation facility with the name of the logical unit. The name translation facility returns the name of the SSCP defined as the owner of the logical unit, as well as the destination network ID and the logical unit's real name in that destination network.

If the VTAM SSCP has a session with the owning SSCP, and dynamic definition of cross-domain resources is allowed, the CDINIT request for the session is sent to the owning SSCP. VTAM automatically creates a CDRSC definition table entry for the logical unit. (You can allow dynamic definition of a resource by coding CDRDYN=YES on the CDRM statement for the VTAM host CDRM and CDRSC=OPT on the CDRM statement for the owning SSCP.)

Alternatives to Predefining Destination LUs

If you have not installed a name translation facility, or if the name translation facility fails to return the name of the owning SSCP, default SSCP selection is tried next. (Defining an alias name translation facility to VTAM is described in "The Alias Name Translation Facility" on page 248.)

Defining Cross – Domain Resources

For SNA network interconnection, a CDRSC major node consists of:

- A VBUILD statement
- Optional NETWORK definition statements
- One or more CDRSC definition statements.

For coding information, see "Defining Cross – Domain Resources" on page 208.

There are no SNA network interconnection considerations for the VBUILD statement. Considerations for the remaining statements are described below.

Cross – Domain Resource: NETWORK

The NETWORK Definition Statement

Name	Definition Statement	Operands
[name]	NETWORK	[NETID = network id]

name

is a 1 – to 8 – character name that 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.

NETID = network id

is an optional 1 – to 8 – character name that identifies the network in which the CDRSC or set of CDRSCs resides. This NETID pertains to the CDRSC or set of CDRSCs defined following the NETWORK statement and before any other NETWORK statement in the same set of definitions.

The NETWORK definition statement names the network in which the logical units, represented by the CDRSC statements, reside. One or more NETWORK statements may appear anywhere after the VBUILD statement. If consecutive NETWORK statements are coded (without intervening CDRSC statements), only the last one will be used.

If no NETWORK statement exists before a CDRSC definition statement, the default NETID of the CDRSC is the network name of the VTAM in which the definition is filed. Such CDRSCs may be resources within VTAM's network or alias names for resources in another network.

If VTAM was not started with a network ID, only CDRSCs for that VTAM network are processed. All others are ignored.

The CDRSC Definition Statement

Name	Definition Statement	Operands
cdrscname	CDRSC	[CDRM = cdrmname] [,ISTATUS = <u>**ACTIVE**</u>|INACTIVE] [,SPAN = (spanname)]

Although no additional operands are needed on the CDRSC statement for SNA network interconnection, additional considerations do apply to the *cdrscname*.

cdrscname
The name fields of all CDRSC definition statements that are active at one time must be unique and must not conflict with other resource names in the same network.

You can define cross – domain resources in three ways:

1. You can predefine the resource in this network using its real logical unit name in the network (either this network or another network) where it resides. Do this by coding a CDRSC statement after the NETWORK statement identifying the network where the real logical unit resides.

   For same – network resources, the CDRM operand defines the SSCP to which session requests for this resource should be routed. If you do not code the CDRM operand, VTAM uses the list of default CDRMs defined in this network's adjacent SSCP table to find the next SSCP to which this request should be routed. When the CDRM is defined but does not own the destination logical unit, the default SSCP list is used to send session setup requests to SSCPs in the list, until the owning SSCP is found or the end of the list is reached. When the owning SSCP is found, VTAM updates the static CDRSC with its name. This action eliminates the need for the network operator to issue a MODIFY CDRM command because the owning SSCP is different from the CDRM name.

   For cross – network resources (defined using the resource's real name), the CDRM operand names the SSCP that owns the resource (the destination SSCP). VTAM uses the destination SSCP name and network ID to select the adjacent SSCP table for the session. When the CDRSC is coded after the NETWORK statement then the CDRM named in the CDRSC statement must own the destination logical unit.

   Note: If this method results in a conflict because a resource in this host network and a cross – network resource have the same name, you will have to use one of the following two methods to define the CDRSC.

2. You can predefine a cross – network resource using an alias name on a CDRSC statement coded in the originating SSCP. However, the next SSCP along the path to the destination SSCP must be a gateway SSCP.

For cross-network resources (defined using the resource's alias name in the host network), the SSCP named by the CDRM operand is not necessarily the SSCP that owns the resource. VTAM does not use this SSCP name to select the adjacent SSCP table. Instead, this SSCP is added to the selected adjacent SSCP list, and it becomes the first SSCP to which the session request is routed.

3. You can allow dynamic definition of a resource by coding CDRDYN=YES on the CDRM statement for the VTAM host CDRM and CDRSC=OPT on the CDRM statement for the SSCP(s) adjacent to this intermediate SSCP. Use this method if resources in the host network will *not* request sessions with a cross-network resource. In this case, the host is responsible for intermediate SSCP services for this logical unit, since intermediate SSCPs do not contain definitions for the CDRSCs.

Defining Adjacent SSCP Tables

The adjacent SSCP table contains lists of SSCPs that may 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 may be activated in each VTAM host. File the definitions in VTAMLST files (or as members in the SYS1.VTAMLST data set).

You can individually activate multiple adjacent SSCP table definitions. Each new set of definitions is added to the internal table used by VTAM. New definitions for destinations already in the internal table replace the current adjacent SSCP list for those destinations, but you cannot deactivate existing adjacent SSCP lists. If a new table replaces an existing one, VTAM issues a message informing the operator.

An adjacent SSCP table is defined using the following definition statements:

- VBUILD TYPE=ADJSSCP
- NETWORK
- CDRM
- ADJCDRM.

These rules describe how to define an adjacent SSCP table.

1. The first statement is a VBUILD statement with TYPE=ADJSSCP.

2. The second statement is a NETWORK or ADJCDRM statement. Not coding a NETWORK statement immediately after the VBUILD statement is the same as coding a NETWORK statement with no NETID operand; this defines a default SSCP list.

3. Consecutive NETWORK statements define the same default SSCP list for each network.

4. VTAM ignores the name fields on the VBUILD and NETWORK statements, but they are used in error messages by the system definition processor.

5. If you code no NETID start option when VTAM is started, only default SSCP lists for this VTAM network are processed. If this is the case, you must allow the NETID operand on the NETWORK statement to default by either omitting it entirely or by giving it a null value.

6. One or more CDRM statements may follow NETWORK statements or ADJCDRM statements. The name field of each CDRM statement names a destination SSCP in the network identified by the preceding NETWORK statement, or, if there is no NETWORK statement, in the network where the table is defined. The CDRM statement is followed by another CDRM statement or an ADJCDRM statement.

7. The CDRM definition statements within the adjacent SSCP table have no required operands, although any operands that are valid on the CDRM statement in a CDRM major node are accepted. VTAM ignores

these operands when they appear in an adjacent SSCP table. Thus, you can define an adjacent SSCP table to reach a new host in an interconnected network simply by adding to the CDRM statements that already exist for that network.

8. Consecutive ADJCDRM definition statements following the CDRM statement(s) provide a list of adjacent SSCP names for the preceding destination SSCP(s) identified on the CDRM statement(s). The name field of each ADJCDRM statement is the name of an adjacent SSCP. Each ADJCDRM statement must have a unique cdrmname. There are no operands on the ADJCDRM statement.

9. The ADJCDRM statement is followed by another ADJCDRM statement, a CDRM statement, or a NETWORK statement (or nothing, if it is the end of the table).

10. If ADJCDRM statements follow a NETWORK or VBUILD statement, without intervening CDRM statements, they comprise a list of default SSCPs. Two types of default SSCP lists are possible depending on the following conditions:

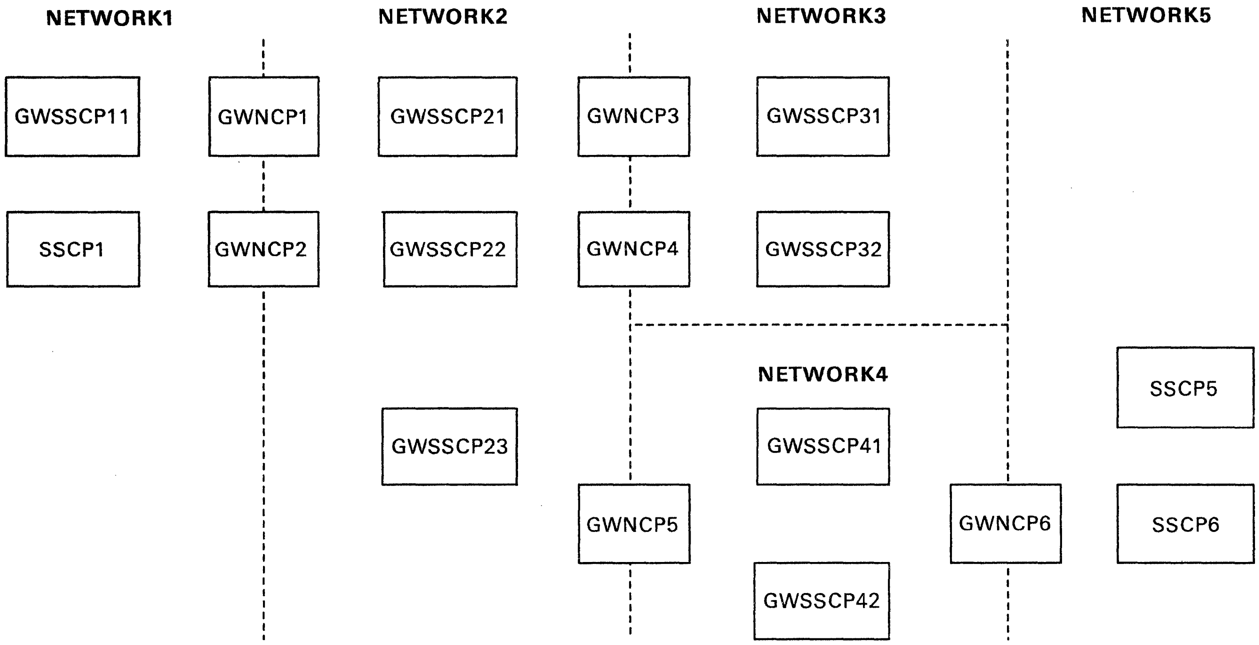
    a. If you code the NETID of another network, the NETID is that of the destination network. The ADJCDRM statements name the SSCPs to use when the destination network ID is known, and the destination SSCP name is not known or does not have a CDRM statement coded for it in the table.

    b. If the NETID is defaulted or is the host network ID, the ADJCDRM statements name the adjacent SSCPs used for sending requests when the destination network ID is not known, or is known to be the host network. In a multiple-network environment, this list includes the default SSCPs for both same-network and cross-network session requests. This same default SSCP list is used if there is no list for a particular destination network.

This section describes using adjacent SSCP tables for interconnected networks. For information on using default SSCP selection in a single network, see "Defining Default SSCP Lists Within an Adjacent SSCP Table" on page 213.

Example of an Adjacent SSCP Table

The sample adjacent SSCP table definitions included here are written for the configuration in Figure 37 on page 277. Keep in mind that this example and those that follow are for illustrative purposes only.



Note: The connections between hosts and NCPs have been omitted for clarity. You can assume that each NCP connects to each host in the networks it joins.

Figure 37. Example of a Multiple – Network Configuration

The adjacent SSCP table definition for gateway SSCP11 (GWSSCP11) is:

```
TABLE1    VBUILD   TYPE=ADJSSCP
NETWORK3  NETWORK  NETID=NETWORK3           destination network
GWSSCP31  CDRM                              destination SSCP
GWSSCP32  CDRM                              destination SSCP
GWSSCP21  ADJCDRM                           adjacent SSCP
GWSSCP22  ADJCDRM                           adjacent SSCP
GWSSCP23  ADJCDRM                           adjacent SSCP
*
NETWORK4  NETWORK  NETID=NETWORK4           destination network
GWSSCP41  CDRM                              destination SSCP
GWSSCP21  ADJCDRM                           adjacent SSCP
GWSSCP22  ADJCDRM                           adjacent SSCP
GWSSCP23  ADJCDRM                           adjacent SSCP
GWSSCP42  CDRM                              destination SSCP
GWSSCP23  ADJCDRM                           adjacent SSCP
GWSSCP22  ADJCDRM                           adjacent SSCP
GWSSCP21  ADJCDRM                           adjacent SSCP
*
NETWORK5  NETWORK  NETID=NETWORK5           destination network
SSCP5     CDRM                              destination SSCP
SSCP6     CDRM                              destination SSCP
GWSSCP23  ADJCDRM                           adjacent SSCP
GWSSCP21  ADJCDRM                           adjacent SSCP
GWSSCP22  ADJCDRM                           adjacent SSCP
```

Figure 38. Adjacent SSCP Table Definition for GWSSCP11

Adjacent SSCP Tables

If GWSSCP11 receives an initiate request for a destination logical unit owned by GWSSCP31 in NETWORK3, and GWSSCP11 knows the destination logical unit's owning SSCP and network, VTAM uses the adjacent SSCP table to determine what route to use for the session setup path. VTAM first looks in the table for the destination network, NETWORK3, and the destination SSCP, GWSSCP31. The adjacent CDRMs given for that destination are GWSSCP21, GWSSCP22, and GWSSCP23. VTAM attempts to establish the session using the adjacent SSCPs in the order they appear in the table.

This example does not include a list of adjacent SSCPs for NETWORK2. The list could be included, but it is not always necessary. A host does not need an adjacent SSCP list for a network, if:

1. The host's SSCP is directly in session with the destination SSCP (that is, there are no intermediate SSCPs on the session path), and

2. The host SSCP knows the destination SSCP name.

In this case, the host SSCP simply sends the request to the destination SSCP. VTAM knows the destination SSCP name once it has been determined from a CDRSC definition or from the alias name translation facility. In this example, GWSSCP11 is directly in session with each SSCP in NETWORK2 and all destination SSCP names are known.

Example of a Default SSCP List

The default SSCP list is used whenever one of these conditions is true:

- The destination network is not known

- The destination network is known, but the destination SSCP name is not available on the session initiation request

- The destination SSCP name is not explicitly defined in the adjacent SSCP table.

The default SSCP list is part of the adjacent SSCP table. You can include a default SSCP list for each network.

The following adjacent SSCP table for GWSSCP21 in NETWORK2 contains default SSCP lists for all the networks in Figure 37 on page 277.

```
TABLE2    VBUILD   TYPE=ADJSSCP
          NETWORK                    unnamed network definition
GWSSCP11  ADJCDRM                    default adjacent SSCP list
GWSSCP31  ADJCDRM                      for all networks
GWSSCP32  ADJCDRM
GWSSCP41  ADJCDRM
GWSSCP42  ADJCDRM
*
          NETWORK  NETID=NETWORK1    destination network
SSCP1     CDRM                       destination SSCP
GWSSCP11  ADJCDRM                    adjacent SSCP for NETWORK1
*
          NETWORK  NETID=NETWORK3    destination network
GWSSCP32  ADJCDRM                    default adjacent SSCP list
GWSSCP31  ADJCDRM                      for NETWORK3
*
          NETWORK  NETID=NETWORK4    destination network
          NETWORK  NETID=NETWORK5    destination network
GWSSCP41  ADJCDRM                    default adjacent SSCP list
GWSSCP42  ADJCDRM                     for NETWORK4 and NETWORK5
*
SSCP5     CDRM                       destination SSCP
GWSSCP42  ADJCDRM                    adjacent SSCP list for
GWSSCP41  ADJCDRM                      for SSCP5 in NETWORK5
```

Figure 39. Adjacent SSCP Table Definition for GWSSCP21

In this table, the default SSCP list for all networks is

GWSSCP11
GWSSCP31
GWSSCP32
GWSSCP41
GWSSCP42

This list is used whenever the destination logical unit's network is not known or when the destination logical unit's network is the same as the host's network, NETWORK2. If the SSCP that sent the session setup request is in the list, it is excluded when the initiate request is routed. This prevents a request from looping. The exclusion of the origin SSCP is within the copy of the default list associated with this session setup. The actual table you defined is unaltered.

The default SSCP list for session requests for NETWORK1 is

GWSSCP11

This list (of one SSCP) is used whenever VTAM knows that NETWORK1 is the destination network, and SSCP1 is the destination SSCP.

The default SSCP list for session requests for NETWORK3 is

GWSSCP32
GWSSCP31

This list is used whenever VTAM knows that NETWORK3 is the destination network but does not know the destination SSCP name.

The default SSCP list for NETWORK4 and NETWORK5 is the same:

GWSSCP41
GWSSCP42

This list is used whenever VTAM knows the destination network to be either NETWORK4 or NETWORK5 but does not know the destination SSCP name.

If the destination SSCP is known to be SSCP5 in NETWORK5, the following adjacent SSCP list is used instead.

GWSSCP42
GWSSCP41

For information on coding a default SSCP list in a single– network configuration, see "Defining Default SSCP Lists Within an Adjacent SSCP Table" on page 213.

Overriding an Adjacent SSCP List

If a list of adjacent SSCPs exists for a destination network and/or SSCP, and VTAM already has an SSCP–SSCP session with the destination SSCP, session setup processing can override the list; that is, that destination SSCP becomes the first choice for the specific session request. This override is temporary and does not change the actual adjacent SSCP table that you defined.

Thus, the default SSCP list, TABLE2, includes the following statements:

```
         NETWORK  NETID=NETWORK3  destination network
GWSSCP32 ADJCDRM                  default adjacent SSCP list
GWSSCP31 ADJCDRM                   for NETWORK3
```

Assume that a cross–network SSCP–SSCP session exists between GWSSCP21 and GWSSCP31. If GWSSCP21 received a cross–network initiate request for a session with a logical unit owned by GWSSCP31 in NETWORK3, the list of adjacent SSCPs in the table—

GWSSCP32
GWSSCP31

—would be reordered to

GWSSCP31
GWSSCP32

for this session setup.

The destination SSCP name, GWSSCP31, is available in the initiate request if the alias name translation facility or a predefined CDRSC were used.

Using SSCPDYN and SSCPORD to Improve Performance

For optimum performance and reliability during session setup, VTAM has the following characteristics.

- VTAM adds a new entry to the adjacent SSCP table whenever it receives a session initiation request from an SSCP that is not already in the table.

- When it looks for an SSCP with which to initiate a session, VTAM scans the adjacent SSCP table in *priority* order, giving preference to:

  - SSCPs for which the most recent session initiation attempt succeeded
  - SSCPs for which no session initiation attempts have been made.

For performance reasons, however, you may wish to change one or both of these characteristics. The SSCPDYN and SSCPORD start options allow you to do so. Use the following table to help you decide how to set SSCPDYN and SSCPORD in your system.

SSCPDYN=YES SSCPORD=PRIORITY	VTAM adds entries dynamically to the adjacent SSCP table. When establishing sessions, VTAM gives priority to SSCPs for which the most recent session initiation attempt succeeded or for which no attempt has been made—even if a more direct path becomes available. This combination of options gives you the greatest flexibility for setting up routes across networks, and, if your adjacent SSCP table is large, it gives you the best performance during session setup.
SSCPDYN=YES SSCPORD=DEFINED	Entries are added dynamically to the adjacent SSCP table, but VTAM always searches the table from top to bottom when it tries to initiate a session. Since the dynamic entries are added at the bottom of the list, they are the last ones tried during session initiation. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions.
SSCPDYN=NO SSCPORD=PRIORITY	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC statement. Thus VTAM can only use the paths you define in the table, and if one of these paths fails VTAM does not attempt to use it again until all the other paths also fail. This combination of options tends to limit the number of available cross-network routes. You may also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application issues a CLSDST PASS macro for a cross-network session.

SSCPDYN=NO SSCPORD=DEFINED	No entries are added to the adjacent SSCP table, except when the owner of a cross-domain resource is defined explicitly by the CDRM operand of the CDRSC statement. VTAM always searches the table from top to bottom when it tries to initiate a session. Again, VTAM can only use the paths you define in the table, but—since VTAM always searches the table from top to bottom—a path that fails can be used again as soon as it is restored. This combination of options results in increased session setup time if the primary paths (those near the top of the list) are not available. But if a secondary path (one nearer the bottom of the table) is used, it does not get priority for subsequent sessions. You may also need to define additional adjacent SSCP tables because VTAM does not build them automatically each time an application issues a CLSDST PASS macro for a cross-network session.

For complete descriptions of these start options, see "The SSCPDYN Start Option" on page 296 and "The SSCPORD Start Option" on page 298.

The VBUILD Statement

Code a VBUILD statement for each adjacent SSCP table.

Name	Definition Statement	Operands
[name]	VBUILD	TYPE=ADJSSCP

name

the *name* is optional.

Adjacent SSCP Table: NETWORK

The NETWORK Definition Statement

The NETWORK statement names a destination network. The NETWORK statement is required for defining an adjacent SSCP table. If you do not code a NETWORK 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 statement to provide a specific default SSCP list for a given destination network. If ADJCDRM statements follow a NETWORK statement, without intervening CDRM statements, they also comprise a list of default SSCPs. See "Defining Default SSCP Lists Within an Adjacent SSCP Table" on page 213 for more information.

Name	Definition Statement	Operands
[name]	NETWORK	[NETID=destination network id]

name
is a 1– to 8–character name that identifies this NETWORK statement in error messages from the system definition processor. To avoid confusion, use the same value for both *name* and the NETID operand.

NETID=destination network id
is the optional 1– to 8–character name of the destination network. Omitting NETID defines a default SSCP list.

The next statement after the NETWORK statement can be a NETWORK, CDRM, or ADJCDRM statement. Consecutive NETWORK statements result in the same adjacent SSCP table for each of the networks.

The CDRM Definition Statement

One or more CDRM statements may follow a NETWORK or ADJCDRM statement. The CDRM statement names a destination SSCP in the network identified on the preceding NETWORK statement.

Name	Definition Statement	Operands
sscpname	CDRM	

sscpname
is the real name of the SSCP. It must match the name assigned to that SSCP in its own network by the SSCPNAME start option or, if SSCPNAME was not coded, by the host CDRM definition statement.

The CDRM 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 statement is part of an adjacent SSCP table.

Note: You can define adjacent SSCP tables to reach a new interconnected network by only adding a few statements to the existing CDRM definitions for that network.

A CDRM statement can be followed by another CDRM statement or an ADJCDRM statement.

Adjacent SSCP Table: ADJCDRM

The ADJCDRM Definition Statement

Consecutive ADJCDRM definition statements following the CDRM statements give the list of adjacent SSCP names for the preceding destination SSCPs.

Name	Definition Statement	Operands
cdrmname	ADJCDRM	

cdrmname
is the name of an adjacent SSCP as known to VTAM. It is the same as the name on the CDRM definition statement that defines the SSCP to VTAM. (An adjacent SSCP may be an SSCP in a non-adjacent network; for example, in a back-to-back configuration. However, an SSCP-SSCP session must have been established.) The CDRM names must be unique among SSCPs that will participate in cross-network sessions.

There are no operands on the ADJCDRM definition statement.

An ADJCDRM definition statement can be followed by another ADJCDRM statement or a CDRM or NETWORK statement.

Defining VTAM Requirements for the NCP

The PCCU Definition Statement

Two of the operands listed in "The PCCU Definition Statement" on page 171 apply specifically to SNA network interconnection:

- GWCTL = ONLY|<u>SHR</u>
- NETID = host network id.

GWCTL = ONLY|<u>SHR</u>
determines whether the host's SSCP is designated to do all the gateway control functions during LU–LU session setup. GWCTL determines the SSCP's role only for the gateway NCP being defined and applies only if the SSCP is on the session setup path.

Notes:

1. *You may code GWCTL = ONLY for only one SSCP within a gateway.*
2. *If the session setup path crosses two gateway SSCPs within a single gateway, either gateway SSCP may be designated to do all the gateway functions. Code GWCTL = ONLY on the PCCU definition statement for the gateway SSCP you choose to do all the gateway functions.*
3. *If the session setup path crosses three gateway SSCPs within a single gateway, only the center gateway SSCP can be designated to do all the gateway functions. Code GWCTL = ONLY on the PCCU definition statement for the center gateway SSCP.*
4. *If the session setup path crosses more than three gateway SSCPs within a single gateway, none can be designated to do all the gateway functions. You must code GWCTL = SHR on the PCCU definition statement for each gateway SSCP in the session setup path.*

Gateway Control Functions: The gateway control functions, designated by the GWCTL operand, apply to LU–LU session setup processing. The functions are:

- To send an RNAA RU to the gateway NCP to assign the pair of alias network addresses for the session
- To send a SETCV RU to the gateway NCP for address translation
- To send a SETCV RU to the gateway NCP specifying the virtual route (VR) list

- To send a SETCV RU to the gateway NCP identifying names to be substituted in BIND.

If you code GWCTL=SHR or let it default, shared control of the gateway NCP is handled as follows:

- The gateway SSCP on the destination logical unit's side of the gateway NCP sends an RNAA RU.

- The gateway SSCP on the destination logical unit's side of the gateway NCP sends a SETCV RU for address translation

- The gateway SSCP on the origin logical unit side of the gateway NCP sends a the SETCV RU for the VR list.

- The gateway SSCP on the origin logical unit side of the gateway NCP sends a SETCV RU for names to be substituted in BIND.

NETID=host network ID

is the 1- to-8 character name of the network of the host represented by this PCCU definition statement. It is used along with the SUBAREA operand to determine which PCCU definition statement the VTAM host should process. This operand is optional.

VTAM uses the following rules to determine which PCCU definition statement to process:

1. The VTAM host processes the first PCCU definition statement whose NETID and SUBAREA operands match VTAM's.

2. If no such PCCU definition statement exists, the VTAM host processes the first one that has the same NETID as VTAM and that does not have a SUBAREA.

3. If a PCCU definition statement as described in 1 or 2 does not exist, the VTAM host processes the first PCCU definition statement without a NETID operand but with a SUBAREA value equal to VTAM's.

4. Finally, if a PCCU definition statement as described in 1, 2, or 3 does not exist, the VTAM host processes the first PCCU statement for which neither NETID nor SUBAREA is coded.

Considerations for VTAM Releases that Do Not Support Interconnection: When processing the PCCU statement, these releases of VTAM ignore the NETID operand:

- MVS All releases before V2R2
- VSE All releases
- VM V3R1.

(In these releases, VTAM issues an informational message when it encounters NETID.)

More than one PCCU 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 statement VTAM finds (regardless of NETID) with that SUBAREA value is processed for that host. For this reason, be careful when providing an NCP definition to one of these releases of VTAM.

If more than one such host (each in a different network) with the same SUBAREA value will process the same set of NCP definitions, you must change the definitions in at least one of the hosts. Delete PCCU statements with matching subareas that do not apply to the host, or change their sequence so that they come after the PCCU statements that do apply.

These releases also ignore the GWCTL operand. (However, VTAM issues an informational message when it encounters GWCTL.) SSCP–SSCP sessions between one or two such VTAMs resolve shared control rules independent of the GWCTL operand.

For complete coding information for PCCU, see "The PCCU Definition Statement" on page 171.

VTAM Requirements for the NCP: BUILD

The BUILD Definition Statement

The BUILD definition statement is described in *NCP – SSP Installation and Resource Definition Guide*. The VTAM – only operands and operands that have VTAM restrictions for SNA network interconnection are described as follows:

COSTAB = class of service table name

This operand is an optional VTAM – only operand. It names a class of service table for the network identified by the NETID operand. The COS table name matches the name of the 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 VR list when the operator issues the command

DISPLAY ROUTE,ORIGIN = ncp subarea node,COSNAME = cos name

NETID = network id

If the BUILD definition statement defines a gateway NCP, and the NETID operand is the same as the gateway SSCP's NETID start option, the gateway SSCP is in the NCP's native network. The BUILD definition statement and the definition statements that follow it up until the first NETWORK definition statement define resources to the native network and are processed by this VTAM. In addition, the SUBAREA operand on the BUILD definition statement is the correct subarea address of the NCP in the gateway SSCP's network. The MAXSUBA operand also applies to the gateway SSCP's network and is consistent with the MAXSUBA value for VTAM.

If NETID is not coded on the BUILD definition statement, the NCP is not a gateway NCP, and all NETWORK definition statements in the NCP definition are ignored.

If VTAM was started without a NETID, it is assumed that the NETID on the BUILD definition statement is the same as VTAM's network ID. Therefore, all the resources defined in the NCP definition are assumed to be owned by this host. If the NCP resources are not to be owned by this host, this NCP should only be activated with a SCOPE = ONLY request. If only connectivity is desired to a channel – attached gateway NCP, define and activate the channel explicitly with a channel attached – data host channel definition.

If the NETID operand does not match the gateway SSCP's NETID (that is, if the gateway SSCP is not in the native network), the SUBAREA is processed for the given NETID when it does not match the gateway SSCP's NETID. The only BUILD definition statement operands that VTAM uses in this case are:

- SUBAREA
- LOADLIB and NEWNAME (the operands needed to load the resource resolution table)
- COSTAB (the operand needed to control the gateway NCP)
- MAXSUBA (applies only to MVS V2R2; VM V3R1).

In addition, since resources attached to the NCP can only be controlled by an SSCP in another network (since NETIDs do not match), everything following the BUILD definition statement is ignored by VTAM *except:*

- The SYSCNTRL definition statement
- The HOST definition statement
- The NETWORK definition statement.

Considerations for VTAM Releases that Do Not Support Interconnection: The following considerations apply to hosts which are running any of these releases of VTAM:

- MVS All releases before V2R2
- VSE All releases
- VM V3R1.

Such a VTAM in a non-native network can activate an NCP if its MAXSUBA and SUBAREA values are compatible with those used within the native network. If the MAXSUBA and SUBAREA values for the non-native network are not compatible with those in the native network, VTAM cannot activate the NCP.

If the values do not match, when the NCP is link-attached to the host, you can delete the NCP major node member from that host's VTAMLST file (or SYS1.VTAMLST data set). The network can still be accessed.

If the NCP is channel-attached to the host, change the SUBAREA and MAXSUBA values in the BUILD definition statement filed in the host to values that are appropriate for its network, if possible. In addition, for a channel-attached NCP, the operands that exist on the first HOST definition statement with a SUBAREA value corresponding to the host's SUBAREA (regardless of the NETID value) are used. If necessary, also change the operands on the HOST definition statement. If the NCP resources are not to be owned by the host, this NCP should only be activated with a SCOPE=ONLY request, and only to activate the NCP major node.

VTAM Requirements for the NCP: HOST

The HOST Definition Statement

The HOST definition statement is described in *NCP−SSP Installation and Resource Definition Guide*. It describes the operands that have VTAM restrictions for SNA network interconnection.

More than one HOST definition statement may exist in the NCP major node. VTAM determines which HOST definition statement applies to the VTAM host.

NETID=network id
is the 1− to 8−character name of the network of the host represented by this HOST definition statement. It is used along with the SUBAREA operand to determine which HOST statement the VTAM host should process. This operand is optional.

VTAM uses the following rules to determine which HOST definition statement to process:

1. The VTAM host processes the first HOST definition statement whose NETID and SUBAREA operands match VTAM's.

2. If no such HOST definition statement exists, the VTAM host processes the first one that has the same NETID as VTAM and that does not have a SUBAREA.

3. If HOST definition statements as described in 1 or 2 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.

4. A HOST definition statement with neither NETID nor SUBAREA coded is assumed to have the default SUBAREA value assigned by the NCP generation process, which is subarea 1.

Defining a Cross−Network Controlling Primary logical unit: Using the LOGAPPL operand, you can define a cross−network controlling primary logical unit for any of the following:

- A local non−SNA device
- A local SNA logical unit
- A switched SNA device.

The effect is the same as if a same−network CDRSC were the controlling primary logical unit, except that the resulting session is cross−network rather than same−network.

For coding information, see the appropriate section in Chapter 5, "Defining the Network to VTAM" on page 69 for defining major nodes and the *NCP−SSP Resource Definition Reference*.

For information on deferred session setup or controlling primary logical units for dial−out resources, see "Deferred Session Setup" on page 249 and "Sessions With Dial−Out Resources" on page 249.

Considerations for VTAM Releases that Do Not Support Interconnection: For a channel–attached NCP, the operands on the first HOST statement with a SUBAREA value corresponding to the host's subarea (regardless of the NETID value) are used. If necessary, change the operands on the HOST definition statement.

This restriction applies only to hosts which are running these releases of VTAM:

- MVS All releases before V2R2
- VSE All releases
- VM V3R1.

VTAM Requirements for the NCP: NETWORK

The NETWORK Definition Statement

The NETWORK definition statement is described in *NCP – SSP Installation and Resource Definition Guide.* The VTAM – only operands and operands that have VTAM restrictions for SNA network interconnection are described here.

You may include more than one NETWORK definition statement in the NCP major node to identify each network attached to a gateway NCP. The native network of the NCP, however, is identified by the NETID operand on the BUILD definition statement. If VTAM was started without a network ID or NETID is not coded on the BUILD definition statement, NETWORK definition statements within the NCP definition are ignored. Otherwise, VTAM uses the following operands:

COSTAB = class of service table name
names a class of service table for the network identified by NETID. This VTAM – only operand is optional and is processed regardless of which network corresponds to the NETWORK definition statement

MAXSUBA = n (for communication with pre – Version 3 nodes only)
is the MAXSUBA start option value used for the network identified by NETID. *n* is the highest subarea value (in decimal) that can be assigned to any pre – Version 3 node in this network. A pre – Version 3 node is either a host processor running a version of VTAM, *or* a communication controller running a version of NCP, for which extended network addressing is not supported. *n* is an integer between 3 and 255. If you code MAXSUBA, the range of its values must be the same for all nodes in the same network. That is, the MAXSUBA values must all use the same number of bits. In networks where there are no pre – Version 3 nodes, you do not need to code this option.

See *Network Program Products Planning* for considerations when coding this operand.

SUBAREA = subarea address
is 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.

NETID = network id
is the 1 – to 8 – character network identifier of the non – native network in which the gateway SSCP resides. If the value for NETID matches the value of VTAM's NETID start option, then 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.

The GWNAU Definition Statement

The GWNAU definition statement is an NCP – only definition statement. It is described in *NCP – SSP Installation and Resource Definition Guide.* However, appropriate coding of this statement determines whether VTAM can activate SSCP – SSCP sessions in some configurations.

The following cases exist where the SSCP that establishes a cross – network session with another SSCP must know that SSCP's alias address within the gateway NCP before it can send any session setup requests for the SSCP – SSCP session:

1. An SSCP can neither send nor receive an ACTCDRM request to establish a session with another SSCP unless it knows both that SSCP's subarea and element addresses. A non – gateway SSCP cannot get the address from the gateway NCP; thus, the non – gateway SSCP depends on the GWNAU definition statement corresponding to the other – network SSCP to supply the alias address. You can code this address on the SUBAREA and ELEMENT operands of the CDRM definition statement that defines the other – network SSCP to the non – gateway host.

2. In a back – to – back gateway configuration, a gateway SSCP must be able to tell its gateway NCP what address represents the other – network SSCP in the second gateway NCP. The address in the GWNAU definition statement for the other – network SSCP in the second gateway NCP is the same as that in the ADJNETSA and ADJNETEL operands of the GWPATH definition statement following the CDRM statement that defines the other – network SSCP to the gateway host.

Sometimes an ACTCDRM request may arrive at a gateway NCP before VTAM has determined the address. In this case, the gateway NCP rejects the request. At the same time, the NCP signals the destination SSCP to set up the session. The NCP knows the address of the destination SSCP only if the SSCP name in the GWNAU definition statement for the alias address to which the ACTCDRM request was sent matches an SSCP name that the NCP received in an ACTPU request. This SSCP name sent in an ACTPU request from a VTAM gateway SSCP is determined by the SSCPNAME start option.

Defining VTAM Start Options

The SSCPID Start Parameter

SSCPID = n

The value of SSCPID must be unique in a network. In addition, it must be unique from those coded for SSCPs in other networks that may be in session with this SSCP. However, more than one SSCP in different networks may have the same SSCPID value if those SSCPs do not have sessions with one another.

Defining SSCPIDs in a multiple-network environment ensures that only a single cross-network SSCP-SSCP session is set up with a given SSCP. This parameter is required.

The NETID Start Option

NETID = network id

is the 1- to 8-character name of the network containing the host.

The network ID allows VTAM to determine which gateway NCP definition statements apply to the gateway host. It is also used by the gateway SSCP when sending and receiving cross-network requests.

Both NETID and SSCPNAME are required in a gateway configuration. If you code one without the other you will, at initialization, be reminded to supply the missing operand if VTAM is to serve as a gateway SSCP.

Code NETID whenever VTAM is participating in cross-network sessions, regardless of whether the SSCP is to be a gateway SSCP. If you do not code NETID, VTAM uses only the subarea value to determine which NCP definition statements apply to the gateway host and considers its network ID to be 8 blanks. If you do not code NETID, VTAM processing of the NCP, adjacent SSCP tables, and the CDRM and CDRSC major nodes is affected. (For details see the sections about defining these major nodes in this chapter and in Chapter 5, "Defining the Network to VTAM" on page 69.) In addition, display output includes NETID information only if you specify NETID when you start VTAM.

The SSCPDYN Start Option

SSCPDYN = <u>YES</u>|NO

determines whether VTAM is to add entries dynamically to the adjacent SSCP table.

If SSCPDYN = YES, the default, 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.

If SSCPDYN=NO, VTAM does not add new entries *unless* it knows the owner of the resource. In this case, VTAM adds the name of the owner to the table. Ownership is determined by the CDRM operand of the CDRSC definition statement, or by session initiation.

Note that, if you set SSCPDYN=NO, you may have to define additional adjacent SSCP tables yourself. When an application issues a CLSDST PASS macro during session setup, VTAM does *not* build an adjacent SSCP table in the application's network. (When SSCPDYN=YES in the application's network, VTAM builds such a table automatically.)

This start option is closely related to SSCPORD. "Using SSCPDYN and SSCPORD to Improve Performance" on page 281 contains information on how you can use them together to get optimum performance.

The SSCPNAME Start Option

SSCPNAME=name
is the 1- to 8-character name of the VTAM SSCP. It is required only if VTAM is to be a gateway SSCP, although it is recommended for all SSCPs.

Both NETID and SSCPNAME are required in a gateway configuration. If you code one without the other you will, at initialization, be reminded to supply the missing operand if VTAM is to serve as a gateway SSCP.

Below are three items to consider when coding the SSCPNAME start option. The first item is a *requirement.* The second and third items are suggestions to improve usability and ease network management.

1. 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.)

2. The SSCPNAME should match the name on the host CDRM statement.

3. The SSCPNAMEs and CDRM names must be unique among SSCPs that will participate in cross-network sessions.

The SSCPNAME is sent to the gateway NCP in the ACTPU request that establishes the SSCP-NCP session and in the request to translate the network address for the SSCP-SSCP session. The gateway NCP uses the name to select the correct GWNAU definition to represent the SSCP.

For more information on the GWNAU definition statement, see *NCP-SSP Installation and Resource Definition Guide.*

Start Options: SSCPORD

The SSCPORD Start Option

SSCPORD = PRIORITY|DEFINED

determines whether VTAM is to search an adjacent SSCP table in priority order (the default) or in the order in which the table is defined.

Priority order means that VTAM gives preference to SSCPs for which the most recent session initiation attempt succeeded, or for which no sessions have been tried. Note that when a CDRM – CDRM session fails, the entry for that CDRM in the table is marked "failed" and cannot be used until all other entries have been tried or the CDRSC is deactivated.

Defined order means that VTAM scans the table in the same order that SSCPs are listed, regardless of whether past session initiation attempts were successful.

This start option is closely related to SSCPDYN. "Using SSCPDYN and SSCPORD to Improve Performance" on page 281 contains information on how you can use them together to get optimum performance.

Chapter 8. Verifying the Installation

This chapter contains procedures you can use to verify that VTAM is installed properly. You need not do every step in the procedures; they are only suggestions of things you can do to "exercise" your newly-installed system.

The command syntax and some of the terminology, are oriented toward MVS systems; some of the material will not apply if you have a VSE or VM system. (VM If you are a VM user, you can also use the procedure in "Verifying the Installation" on page 45 to verify the initial installation.)

When you follow the procedures in this chapter, be sure that ISTATUS=INACTIVE for all resources. Lowercase names in commands are placeholders; you should substitute the actual names of resources in your system.

If these procedures do not yield the expected results, double-check the definition statements for the resources that are affected. If this does not solve the problem, refer to the *VTAM Diagnosis Guide* for help.

Verifying Single-Domain Systems

Now that you have entered all statements coded during the installation phase, you can exercise the components together as a network.

To verify a single-domain system, follow these steps:

1. Start VTAM and start the appropriate trace facility (such as MVS GTF or VM CPTRAP).

2. Activate a channel-attachment major node, begin tracing, and activate lines and peripheral physical units.

3. Activate an NCP, begin tracing and intensive mode error recording, and verify that you can dump the NCP.

4. Start NetView (or start NCCF and NLDM) if it is installed.

5. Activate a switched major node and begin tracing for the physical units.

6. Transmit the SNA terminal subsystem program from the host to a peripheral physical unit.

7. Start the SNA terminal subsystem program at the physical unit.

8. Start the VTAM or host subsystem application programs.

9. Start VTAM traces and activate the logical unit(s) at the peripheral physical unit.

10. Verify peripheral terminals and links.

11. Verify the route structure.

12. Verify sessions between application programs and logical units.

13. Activate and verify additional physical units and NCPs.

14. Simulate normal operation with all physical units.

15. Try backup and recovery procedures.

16. Halt VTAM.

Step 1: Start VTAM and Start the Trace Facility

For example, enter S NET at the VTAM operator's console where NET is a cataloged procedure defined for starting VTAM. For a sample start procedure, see Figure 8 on page 29.

If you have placed PROMPT in the ATCSTR00 start parameter list, VTAM prompts the operator to give any overriding parameters.

If there is an initial configuration list, all the major nodes listed will be activated. This can include any VTAM application program. If any major nodes in this list define NCP major nodes or channel–attachment major nodes, the beginning of the list must have the sets of path definition statements that define the routes to these subarea nodes.

MVS VTAM uses the generalized trace facility (GTF) to record VTAM trace records. To start GTF, enter

S gtfprocname

Refer to:

- *VTAM Operation*
- *OS/VS2 System Programming Library: Service Aids.*

VM VTAM uses CPTRAP to record VTAM trace records. Refer to the *VTAM Diagnosis Guide* for information on how to start traces in a VM environment.

Note: If you plan to interconnect networks, start all VTAM Version 3 hosts with NETID and SSCPNAME start options. This is required for SSCPs that will act as gateway SSCPs.

Step 2: Activate a Channel Attachment Major Node, Begin Tracing, and Activate Lines and Peripheral PUs

To activate the major node, enter:

V NET,ACT,ID=major node name

To activate I/O trace to all nodes defined under the channel–attachment major node, enter:

F NET,TRACE,ID=major node name,E,TYPE=IO

After you initiate the trace, enter the following to activate each nonswitched line and PU:

V NET,ACT,ID=linename

V NET,ACT,ID=puname

Step 3: Activate an NCP, Begin Tracing and Intensive Mode Recording, and Verify that You Can Dump the NCP

You should detect errors as early as possible when you verify the elements of the network. VTAM traces and the intensive mode error recording facility are useful now. (The intensive mode error recording facility records both temporary and permanent errors from a physical unit.)

You should also verify, for future debugging, that you can dump the NCP. You can, if you wish, dump the NCP now. Refer to "Step 2: Dump the NCP and Print the Dump" on page 311.

You can use the following sequence to begin the VTAM internal and buffer traces:

F NET,TRACE,TYPE=VTAM,OPTIONS=ALL,MODE=EXT

F NET,TRACE,ID=VTAM,TYPE=BUF

F NET,TRACE,ID=ISTPUS,TYPE=BUF

To verify that you can dump the NCP, make sure that you have correctly coded the following operands on the NCP PCCU definition statement:

- DUMPDS should name a file to receive the dump records. For more information about defining an NCP dump data set, see "NCP Dump Data Set" on page 25.
- DUMPSTA should name an adjacent link station through which you will dump the NCP.

For information on coding these operands, see "Defining Network Control Program (NCP) Major Nodes" on page 169.

To activate path definition statements that define routes to NCPs, enter:

V NET,ACT,ID=path set name

To activate a channel–attached NCP, enter:

V NET,ACT,ID=ncpname

To start tracing of I/O by the NCP, enter:

F NET,TRACE,ID=ncpname,TYPE=IO

To activate the I/O trace to the physical unit, enter:

F NET,TRACE,ID=puname,TYPE=IO

to activate the nonswitched line and the physical unit, enter:

V NET,ACT,ID=linename

V NET,ACT,ID=puname

To begin intensive mode error recording for the physical unit, enter:

F NET,IMR,ID=puname,OPT=ACT,RECLIM=m

where *m* is the maximum number of temporary errors to be recorded.

Step 4: Activate NetView (or NCCF and NLDM)

If you have NetView (or the NCCF and NLDM program products) installed, activate them. Those products' manuals contain verification procedures.

Step 5: Activate a Switched Major Node and Begin Tracing for the PUs

To activate a switched major node, defining switched resources to VTAM, enter:

V NET,ACT,ID=switched major node name

To begin VTAM tracing of I/O to the physical unit, enter:

F NET,TRACE,ID=puname,TYPE=IO

To activate a dial–in line and put it in answer mode, enter:

V NET,ACT,ANS=ON,ID=linename

To activate the physical unit, allowing VTAM to accept a dial−in request, enter:

V NET,ACT,ID=puname

When the device dials in, its connection to VTAM is established.

For a physical unit with dial−out capability, VTAM dials out when a request is made to establish a session with one of its logical units. Refer to *VTAM Operation* for more information.

Step 6: Transmit the SNA Terminal Subsystem Program from the Host to a Peripheral PU

Before verifying communication between the host and the SNA terminal subsystem program, you may prefer to verify communication between an existing application program in the host and a nonprogrammable device, like a 3270 or 3767. Refer to *VTAM Operation* and the SNA terminal subsystem publications for details.

Step 7: Start the SNA Terminal Subsystem Program at the PU

The SNA terminal subsystem publications contain instructions for this step.

Step 8: Start the VTAM or Host Subsystem Application Programs

For each application program, enter:

START applname

Step 9: Start VTAM Traces and Activate the LU(s) at the Peripheral PU

To begin tracing, enter:

F NET,TRACE,ID=luname,TYPE=IO

F NET,TRACE,ID=luname,TYPE=BUF

To activate each logical unit, enter:

V NET,ACT,ID=luname

Step 10: Verify Peripheral Terminals and Links

As you activate resources of the network, you can apply the logical unit connection and link level 2 verification procedures.

Verifying Logical Unit Connection

To verify the session between the logical unit and its owning SSCP, from the terminal logical unit enter:

IBMTEST n,data

Where *n* is a number from 1 through 255 (with a default of 10), and *data* is a string of up to 255 EBCDIC characters. The SSCP will return the data *n* times, each time preceded by the word IBMECHO.

The IBMTEST command does not verify that the logical unit can establish a session with an application program; it only verifies that the SSCP-LU session is working properly.

Verifying a Nonswitched SDLC Link

You can verify a nonswitched SDLC link between:

- A communication controller and a cluster controller
- Two communication controllers.

To verify the link, first deactivate the link station or physical unit through the SSCP that owns the link, then enter:

F NET,LL2,ID=station-name,...

You can also add data to send over the link and decide how often it should be sent. You will receive:

- The number of test frames transmitted by the host or communication controller
- The total number of test frames returned to the host or communication controller
- The number of test frames returned that were without error

Step 11: Verify the Route Structure

Enter:

D NET,ROUTE,DESTSUB=n,TEST=YES

This will identify and verify available routes going to destination subarea *n*. You can display and verify:

- All explicit routes (ERs) to the destination subarea (ER=ALL)
- All ERs to the destination subarea that have a specific number (ER=n)
- All virtual routes (VRs) to the subarea that are named in a COS table entry (COSNAME=n)

- All VRs that have a specific number (VR = n)
- All the ERs that are associated with the VR you name.

Step 12: Verify Sessions between Application Programs and LUs

Once logical unit has been activated, you should verify it in session with an application program. You should verify:

- Session – initiation procedures, using different logons (if any)
- Message transfer between the application program and the logical unit, including all types of data traffic
- Session termination, using different logoffs (if any)
- Normal error handling, such as conditions that require operator intervention.

Step 13: Activate and Verify Additional PUs and NCPs

Do "Step 2: Activate a Channel Attachment Major Node, Begin Tracing, and Activate Lines and Peripheral PUs" on page 301 through "Step 9: Start VTAM Traces and Activate the LU(s) at the Peripheral PU" on page 303 for additional physical units and logical units in the system. Verify one physical unit at a time.

To activate a local SNA major node, enter:

V NET,ACT,ID = local SNA major node name

To begin tracing, enter:

F NET,TRACE,ID = puname,TYPE = IO

To activate the physical unit, enter:

V NET,ACT,ID = puname,U = channel device address

Note: The U operand is not needed if there is a CUADDR operand in the PU statement in the local definition.

To activate a local non – SNA 3270 major node, enter:

V NET,ACT,ID = major node name

To begin tracing, enter:

F NET,TRACE,TYPE = IO,ID = terminal name

To activate the 3270 controller, enter:

V NET,ACT,ID = terminal name

Step 14: Simulate Normal Operation with All PUs

Do the previous step, using all or several physical units at the same time. In other words, simulate normal operation.

Step 15: Try Backup and Recovery Procedures

Simulate error conditions in the domain. Try all planned backup and recovery procedures to verify their effectiveness.

Try each of the methods for switched network backup. You will find information about line backup using dynamic reconfiguration, and switched network backup with switched major nodes, in *Network Program Products Planning*.

Step 16: Halt VTAM

Using the VARY INACT operator command, deactivate each major and minor node in the network. Halt VTAM with the HALT command.

It is good idea to try all the different options in each command, and all the possible situations. For example, in deactivating an NCP, you might:

- Deactivate any link-attached NCP alone.
- Deactivate any link-attached NCPs and then the channel-attached NCPs to which they are attached.
- Deactivate each channel-attached NCP. Verify that the operator receives route failure notification for routes to NCPs that are link-attached to the channel-attached NCPs. (If you have activated a channel-attached NCP, its link-attached NCPs should be active.)
- Issue VARY INACT commands using the I, F, R, and CDLINK=ACT operands.

Verifying Multiple-Domain Systems

After you finish the steps in "Verifying Single-Domain Systems" on page 299 for each domain in the network, do the steps in this section. You can make a list of the most important normal and backup operations in your system, and verify each of them one at a time. Then verify the system over a length of time, as in normal operation.

To verify a multiple – domain system, follow these steps:

1. Activate cross – domain paths, links, cross – domain resource managers (CDRMs), and cross – domain resources (CDRSCs).

2. Verify cross – domain sessions by exchanging requests across domains between terminals and hosts and between application programs.

3. Verify NetView (or NCCF and NLDM) if it is installed.

4. Verify host backup using a channel – attached shared NCP.

5. Verify host backup over a cross – domain link.

6. Verify automatic SSCP – SSCP session restart.

7. Halt VTAM in each domain.

Step 1: Activate Cross – Domain Paths, Links, CDRMs, and CDRSCs

Besides the resource activation you did in "Verifying Single – Domain Systems" on page 299, you must activate at each host:

- Cross – domain links (LINE and PU link station minor nodes) in the NCP and channel – attachment major nodes

- The VTAM path tables

- CDRM and CDRSC major and minor nodes.

Activate PATH statements in the VTAM hosts first. To do so, enter:

```
V NET,ACT,ID=path set name
```

For an SSCP – SSCP session to be established:

- *The host CDRMs in both domains must be active.*

- *Routes between the SSCPs must be active.*

- *In one host, the minor node CDRM name for the other host must be activated.*

A message at each host confirms that the session has been established.

Step 2: Verify Cross – Domain Sessions

Having activated cross – domain definitions, try to establish a session between a terminal in one domain and an application program in another domain.

Verify cross-domain sessions between application programs by starting an application program in each host and having one establish a session with the other.

Step 3: Verify NetView (or NCCF and NLDM)

If you have NetView (or the NCCF and NLDM program products) in your system, it should already be active. The manuals for those products contain procedures for verifying them in a multiple-domain network.

Step 4: Verify Host Backup Using a Channel-Attached Shared NCP

If you want one host to back up another host that shares a channel-attached NCP, simulate the failure of the main host. (Resources to be backed up must already be defined in the NCP definition major node for the backup host.)

You can do this by having the main host issue VARY NET,INACT,ID=cua-L (where *cua-L* is the name of a channel link station leading to the shared NCP) for the NCP. The backup host should issue the required activation commands and then communicate successfully with the acquired resources.

Step 5: Verify Host Backup over a Cross-Domain Link

Do the previous step from a backup host attached by an SDLC link to the NCP owned by the main host. To simulate a host failure, deactivate all channel link stations at the main host that are in routes leading to the NCP.

Step 6: Verify Automatic SSCP-SSCP Session Restart

Ensure that each SSCP-SSCP session restarts automatically after a failure when both SSCPs have RECOVERY=YES coded (or defaulted) in their CDRM major node definitions.

VM SSCP-SSCP session restart does not apply to V3R1.

Step 7: Halt VTAM in Each Domain

Using HALT commands, ensure that each domain halts successfully.

MVS VM Verifying Interconnected SNA Networks

This section is for verifying SNA interconnected networks only.

After you finish the steps in "Verifying Single-Domain Systems" on page 299 and "Verifying Multiple-Domain Systems" on page 306 for each network, do the steps in this section.

To verify SNA interconnected networks, follow these steps:

1. Activate each gateway SSCP and gateway NCP.
2. Activate major nodes required for cross-network sessions.
3. Establish required cross-network SSCP-SSCP sessions.
4. Verify cross-network sessions by exchanging requests across networks between terminals and hosts and between application programs.
5. Verify termination processing when an SSCP-SSCP or SSCP-gateway NCP session is deactivated.
6. Try backup and recovery procedures.
7. Verify NetView (or NCCF and NLDM) if it is installed.
8. Halt VTAM in each domain in each network.

Most operational functions done in this section are very similar to the ones you did in "Verifying Multiple-Domain Systems" on page 306.

Step 1: Activate Each Gateway SSCP and Gateway NCP

Start each gateway SSCP that is used exclusively for interconnecting networks. (All other SSCPs should already be started.) The procedure is the same as for starting any other SSCP except that the NETID and SSCPNAME start options are required.

At each gateway SSCP that controls a gateway NCP, activate the gateway NCP if it is not already active. This will give the gateway SSCP information about the networks sharing that gateway NCP.

Step 2: Activate Major Nodes Required for Cross-Network Sessions

At each host that contains an SSCP that will have a session with an SSCP in another network, activate CDRM major nodes that contain CDRM statements for cross-network SSCPs.

If you have predefined cross-network resources at a host, activate the CDRSC major nodes that define cross-network resources.

Activate adjacent SSCP tables in the gateway hosts.

If name translation is required, you must activate the alias name translation facility defined in the ALIASAPL APPL definition statement (or a user-written name translation facility).

All these activations are done in addition to the ones you did in the single-domain and multiple-domain procedures.

Step 3: Establish Required Cross – Network SSCP – SSCP Sessions

For each host that contains an SSCP that will have a session with an SSCP in another network, activate the CDRM minor nodes (within the CDRM major nodes) that represent cross – network or additional cross – domain SSCPs with which the host requires a session. These SSCP – SSCP sessions are necessary for cross – network LU – LU sessions to be established.

Step 4: Verify Cross – Network Sessions

Establish an LU – LU session between a terminal or application program in one network and an application program in another network.

Since a session may cross several networks, more than two SSCPs may be needed to establish the session. Therefore, you should be familiar with the configuration and establish LU – LU sessions that represent various session possibilities.

Consider using a large number of LU – LU sessions. You can determine that number by estimating the average number of cross – network sessions through a specific gateway NCP. This is done to assure the system programmer that the gateway NCP(s) will be able to handle the expected number of cross – network sessions.

If you have alternate gateway paths defined, establish LU – LU sessions on them.

If you have alternate paths to other networks, issue a series of Activate and Deactivate commands for the CDRMs named in the ADJSSCP statements in the adjacent SSCP tables that represent gateway SSCPs. This procedure forces LU – LU sessions to use alternate paths for session setup and will show you which adjacent SSCP provides the most efficient session setup.

Step 5: Verify Termination Processing

Terminate various cross – network SSCP – SSCP sessions and SSCP – gateway NCP sessions. Ensure that active cross – network LU – LU sessions continue without the cross – network SSCP – SSCP session. Ensure, also, that cross – network LU – LU sessions that use an alternate gateway NCP (not the gateway NCP whose session with the owning SSCP has been deactivated) continue.

Step 6: Try Backup and Recovery Procedures

Simulate error conditions in the interconnected networks. Verify all cross – network backup and recovery procedures.

Step 7: Verify NetView (or NCCF and NLDM)

If you have NetView (or the NCCF and NLDM program products) in your system, ensure that all necessary NLDM – NLDM sessions can be established. Ensure that correct NCCF definitions have been provided for cross – network sessions. Determine that your domain's NLDM has defined proper authorization for viewing configuration data from NLDMs in other domains.

Verify that the NLDM display panels contain accurate information about the cross – network LU – LU sessions that have been activated.

Exercise the trace functions of NLDM. See *NetView Diagnosis Guide* or *NLDM Installation and Operation* for a discussion of these functions.

Step 8: Halt VTAM in Each Domain in Each Network

Using HALT commands, make sure that each VTAM halts successfully.

Note: If you issue the HALT command at a gateway host that is currently controlling a gateway NCP, that gateway NCP is deactivated. Any cross – network SSCP sessions with the gateway SSCP are also terminated and cross – network LU – LU sessions supported by the gateway SSCP are ended.

Verifying Diagnostic Procedures

Even if you complete all the procedures without encountering errors, it is a good idea to print trace and error records that have been gathered. It is also useful at the end of the installation process to dump an NCP and print the dump.

Step 1: Print Trace and Error Records

Follow the instructions in the *VTAM Diagnosis Guide* to print trace and error records.

Step 2: Dump the NCP and Print the Dump

By dumping the NCP and printing it, you can see how the dump is formatted and what information it contains. This helps you make better use of it during subsequent debugging. You can dump the NCP and print it any time during the procedure in "Verifying Single – Domain Systems" on page 299. (You cannot, however, dump an NCP attached by a channel to a data host.)

To dump the NCP, enter:

```
MODIFY NET,DUMP,ID=ncpname
```

This puts the NCP dump records in the file named in the DUMPDS operand in the PCCU definition statement.

If the NCP fails during any of the procedures and AUTODMP is coded in the PCCU definition statement, you are asked whether you want to dump the NCP. Reply YES or NO to the message and, if YES, print the dump.

For more information on how to dump the NCP, refer to the *NCP/SSP Diagnosis Guide.*

Appendix A. Quick Reference for VTAM Definition Statements

The following quick reference chart lists VTAM definition statements and their operands.

This list does not include any NCP definition statements.

Name	Definition Statement	Operands	Restrictions
name	ADD	TO=resource name	
cdrmname	ADJCDRM		
name	APPL	[ACBNAME=acbname] [,AUTH=([ACQ|**NOACQ**] [,CNM|**NOCNM**] [,PASS|**NOPASS**] [,PPO|SPO|**NOPO**] [,TSO|**NOTSO**] MVS [,**VPACE**|NVPACE])] [,AUTHEXIT=(YES|**NO**)] VM [,DLOGMOD=default logmode entry] [,EAS=n|**491**] [,ENCR=SEL|REQD|OPT|**NONE**] MVS [,HAVAIL=YES|**NO**] MVS/XA [,MAXPVT=**0**|n|nK|nM] MVS VM [,MODETAB=logon mode table name] [,PARSESS=YES|**NO**] [,PRTCT=password] [,SONSCIP=YES|**NO**] [,SPAN=(spanname)] [,SRBEXIT=YES|**NO**] MVS [,SSCPFM=USSNOP|**USSPOI**] [,USSTAB=name] [,VPACING=n|**0**] [,VTAMFRR=YES|**NO**] MVS	AUTHEXIT applies to V3R1 only. VM EAS default is 404 in V3R1.
cdrmname or sscpname	CDRM	[CDRDYN=YES|**NO**] [,CDRSC=OPT|**REQ**] [,ELEMENT=n|**1**] [,ISTATUS=**ACTIVE**|INACTIVE] [,RECOVERY=**YES**|NO] [,SPAN=(spanname)] [,SUBAREA=n] [,VPACING=n|0|**63**]	"sscpname" is for SNA network interconnection only

Name	Definition Statement	Operands	Restrictions
cdrscname	CDRSC	[CDRM=cdrmname] [,ISTATUS=**ACTIVE**|INACTIVE] [,SPAN=(spanname)]	
name	CLUSTER	GPOLL=char [,CUTYPE=**3271**|3275] [,DLOGMOD=default logmode entry] [,FEATUR2=([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,SPAN=spanname] [,TERM=3275|3277|3284|3286] [,USSTAB=USS definition table]	VSE VM
[name]	DELETE	FROM=resource name	
name	GROUP (BSC line)	LNCTL=BSC [,CUTYPE=**3271**|3275] [,DLOGMOD=default logmode entry] [,FEATUR2=([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,RETRIES=n|**7**] [,SERVLIM=n|**4**] [,SPAN=(spanname)] [,TERM=3275|3277|3284|3286] [,USSTAB=USS definition table]	VSE VM
name	GROUP (CTCA)	LNCTL=CTCA [,DELAY=time|0|**.100**] [,ISTATUS=**ACTIVE**|INACTIVE] [,MAXBFRU=([norm|**10**][,max|**norm**])] [,MIH=YES|**NO**] [,PUTYPE=**4**] [,REPLYTO=time|**3.0**] [,SPAN=(spanname)]	
name	GROUP (for a channel-adapter NCP)	LNCTL=NCP [,CHANCON=**COND**|UNCOND] [,ISTATUS=**ACTIVE**|INACTIVE] [,MAXBFRU=n] [,MAXDATA=size|**65535**] [,PUTYPE=**4**] [,SPAN=(spanname)]	VM applies only to V3R1.1.

<table>
<tr><th>Name</th><th>Definition Statement</th><th>Operands</th><th>Restrictions</th></tr>
<tr><td>name</td><td>GROUP
(nonswitched
SDLC line)</td><td>LNCTL=SDLC
[,ACTIVTO=t]
[,DIAL=NO]
[,DISCNT=([YES|NO] [,F|NF])]
[,DLOGMOD=default logmode entry]
[,ISTATUS=ACTIVE|INACTIVE]
[,LOGAPPL=application program name]
[,LOGTAB=interpret table name]
[,MAXBFRU=([norm|1|2] [,max|2|8])]
[,MAXDATA=size|261|265]
[,MAXOUT=n|1]
[,MODETAB=logon mode table name]
[,PACING=n|0|1]
[,PASSLIM=n|maxout]
[,PAUSE=t|0.1]
[,PUTYPE=1|2|4|5]
[,REPLYTO=n|1]
[,RETRIES=n|7]
[,SERVLIM=n|4]
[,SPAN=(spanname)]
[,SSCPFM=FSS|USSSCS]
[,USSTAB=USS definition table]
[,VPACING=n|0|2]</td><td>VSE VM</td></tr>
<tr><td>name</td><td>GROUP
(switched
SDLC line)</td><td>LNCTL=SDLC
,DIAL=YES
[,ACTIVTO=t]
[,ANSWER=ON|OFF]
[,AUTODL=YES|NO] VSE
[,CALL=IN|OUT|INOUT]
[,DIALNO=telephone number] VSE
[,ISTATUS=ACTIVE|INACTIVE]
[,MAXBFRU=([norm|1|2] [,max|2|8])]
[,MAXLU=n|2]
[,PAUSE=t|0.1]
[,REPLYTO=t|1]
[,RETRIES=n|7]
[,RETRYTO=t|12] VSE
[,SERVLIM=n|4]
[,SHOLD=NO|(free,npoll)] VSE
[,SPAN=spanname]
[,X21SW=YES|NO]</td><td>VSE VM</td></tr>
<tr><td>[name]</td><td>GWPATH</td><td>[ADJNET=adjacent network ID]
[,ADJNETEL=adjacent network element|1]
[,ADJNETSA=adjacent network subarea]
[,ELEMENT=element address|1]
[,GWN=gateway NCP name]
[,SUBAREA=subarea address]</td><td>SNA network interconnection only</td></tr>
<tr><td>[name]</td><td>LBUILD</td><td>[CONFGDS=name]
[,CONFGPW=password]</td><td></td></tr>
</table>

Name	Definition Statement	Operands	Restrictions
name	LINE (BSC line)	[ADDRESS = channel unit address|**030**] [,CUTYPE = 3271|3275] [,DLOGMOD = default logmode entry] [,FEATUR2 = (**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])] [,ISTATUS = **ACTIVE**|INACTIVE] [,LOGAPPL = application program name] [,LOGTAB = interpret table name] [,MODETAB = logon mode table name] [,RETRIES = n|**7**] [,SERVLIM = n|**4**] [,SPAN = (spanname)] [,TERM = 3275|3277|3284|3286] [,USSTAB = USS definition table]	VSE VM
name	LINE (CTCA)	[ADDRESS = channel unit address] [,DELAY = time|0|**.100**] [,ISTATUS = **ACTIVE**|INACTIVE] [,MAXBFRU = ([norm|**10**][,max|norm])] [,MIH = YES|**NO**] [,PUTYPE = **4**] [,SPAN = (spanname)]	
name	LINE (link to a channel-adapter NCP)	MAXBFRU = n [,ADDRESS = channel unit address] [,CHANCON = **COND**|UNCOND] [,ISTATUS = **ACTIVE**|INACTIVE] [,MAXDATA = n|**65535**] [,PUTYPE = **4**] [,SPAN = (spanname)]	VM applies only to V3R1.1.
name	LINE (nonswitched SDLC line)	[ADDRESS = channel unit address|**030**] [,ACTIVTO = t] [,DISCNT = ([YES|**NO**] [,**F**|NF])] [,DLOGMOD = default logmode entry] [,ISTATUS = **ACTIVE**|INACTIVE] [,LOGAPPL = application program name] [,LOGTAB = interpret table name] [,MAXBFRU = ([norm|**1**|**2**] [,max|**2**|**8**])] [,MAXDATA = size|**261**|**265**] [,MAXOUT = n|**1**] [,MODETAB = logon mode table name] [,PACING = n|0|**1**] [,PASSLIM = n|**maxout**] [,PAUSE = t|**0.1**] [,PUTYPE = 1|**2**|4|5] [,REPLYTO = n|**1**] [,RETRIES = n|**7**] [,SERVLIM = n|**4**] [,SPAN = spanname] [,SSCPFM = **FSS**|USSSCS] [,USSTAB = USS definition table] [,VPACING = n|0|**2**]	VSE VM

Name	Definition Statement	Operands	Restrictions
name	LINE (switched SDLC line)	[ACTIVTO=t] [,ADDRESS=channel unit address|**030**] [,ANSWER=**ON**|OFF] [,AUTO=address] [,AUTODL=**YES**|NO] VSE [,CALL=**IN**|OUT|INOUT] [,ISTATUS=**ACTIVE**|INACTIVE] [,MAXBFRU=([norm|**1**] [,max|**2**])] [,MAXLU=n|**2**] [,PAUSE=t|**0.1**] [,REPLYTO=t|**1**] [,RETRIES=n|**7**] [,RETRYTO=t|**12**] VSE [,SERVLIM=n|**4**] [,SPAN=spanname]	VSE VM
name	LOCAL	CUADDR=channel device address ,TERM=3277|3284|3286 [,DLOGMOD=default logmode entry] [,FEATUR2=([EDATS|**NOEDATS**] [,**MODEL1**|MODEL2])] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,SPAN=(spanname)] [,USSTAB=USS definition table name]	
name	LU (local)	LOCADDR=n [,DLOGMOD=default logmode entry] [,ENCR=REQD|SEL|OPT|**NONE**] MVS [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,PACING=n|0|**1**] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS] [,USSTAB=USS definition table name] [,VPACING=n|0|**1**]	
name	LU (nonswitched SDLC line)	LOCADDR=n [,DLOGMOD=default logmode entry] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,PACING=n|0|**1**] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS] [,USSTAB=USS definition table name] [,VPACING=n|0|**2**]	VSE VM

Name	Definition Statement	Operands	Restrictions
name	LU (switched)	LOCADDR=n [,BATCH=YES|**NO**] [,DLOGMOD=default logon mode table entry] [,ENCR=REQD|SEL|OPT|**NONE**] MVS [,FEATUR2=LOWERCSE|**DUALCSE**] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MODETAB=logon mode table name] [,PACING=n|0|**1**] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS|USSNTO| USS3780|USS3270|USS3275] [,TERM=terminal type] [,USSTAB=USS definition table name] [,VPACING=n|0|**2**]	
[name]	NETWORK	[NETID=network id]	SNA network interconnection only
[name]	PATH	DESTSA=n|(n1,n2,n3,...) [,ER0=(adjsub[,tg#])] [,ER1=(adjsub[,tg#])] . . . [,ER7=(adjsub[,tg#])] [,VR0=er#] [,VR1=er#] . . . [,VR7=er#] [,VRPWS00=(min#,max#)] [,VRPWS01=(min#,max#)] [,VRPWS02=(min#,max#)] [,VRPWS10=(min#,max#)] [,VRPWS11=(min#,max#)] [,VRPWS12=(min#,max#)] [,VRPWS20=(min#,max#)] . . . [,VRPWS72=(min#,max#)]	
[name]	PATH (switched)	DIALNO=telephone number|LINENM=linename [,GID=n] [,GRPNM=groupname] [,PID=n] [,REDIAL=n|**3**] [,SHOLD=YES|**NO**] VSE [,USE=**YES**|NO]	

Name	Definition Statement	Operands	Restrictions
name	PU (CTCA)	[,DELAY=time|0|**.100**] [,ISTATUS=**ACTIVE**|INACTIVE] [,PUTYPE=**4**] [,SPAN=(spanname)]	
name	PU (local)	[CUADDR=channel device address] [,DISCNT=([YES|**NO**][,**F**|NF])] [,DLOGMOD=default logmode entry] [,ENCR=REQD|SEL|OPT|**NONE**] MVS [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MAXBFRU=n|**1**] [,MODETAB=logon mode table name] [,PACING=n|0|**1**] [,PUTYPE=**2**] [,SECNET=YES|**NO**] [,SPAN=(spanname)] [,SSCPFM=**FSS**|USSSCS] [,USSTAB=USS definition table name] [,VPACING=n|0|**1**]	
name	PU (for an NCP)	[,CHANCON=**COND**|UNCOND] [,ISTATUS=**ACTIVE**|INACTIVE] [,MAXDATA=n|**65535**] [,PUTYPE=**4**] [,SPAN=(spanname)]	VM applies only to V3R1.1.
name	PU (nonswitched SDLC line)	ADDR=char [,DISCNT=([YES|**NO**] [,**F**|NF])] [,DLOGMOD=default logmode entry] [,ISTATUS=**ACTIVE**|INACTIVE] [,LOGAPPL=application program name] [,LOGTAB=interpret table name] [,MAXDATA=size|**261**|**265**] [,MAXOUT=n|**1**] [,MODETAB=logon mode table name] [,PACING=n|0|**1**] [,PASSLIM=n|**maxout**] [,PUTYPE=1|**2**|4|5] [,RETRIES=n|**7**] [,SPAN=spanname] [,SSCPFM=**FSS**|USSSCS] [,SUBAREA=n] [,TADDR=chars|**C1**] [,USSTAB=USS definition table] [,VPACING=n|0|**2**]	VSE VM

Name	Definition Statement	Operands	Restrictions
name	PU (switched)	ADDR = station address ,IDBLK = identification block ,IDNUM = identification number [,BATCH = YES|**NO**] [,DISCNT = (YES|**NO**][,**F**|NF])] [,DLOGMOD = default logmode entry] [,ENCR = REQD|SEL|OPT|**NONE**] MVS [,FEATUR2 = LOWERCSE|**DUALCSE**] [,IRETRY = YES|**NO**] [,ISTATUS = **ACTIVE**|INACTIVE] [,LOGAPPL = application program name] [,LOGTAB = interpret table name] [,MAXDATA = size|**261**|**265**] [,MAXOUT = n|**1**] [,MAXPATH = n|**0**] [,MODETAB = logon mode table name] [,PACING = n|0|**1**] [,PASSLIM = n|**1**] [,PUTYPE = n|**2**] [,SECNET = YES|**NO**] [,SPAN = (spanname)] [,SSCPFM = **FSS**|USSSCS|USSNTO| USS3780|USS3270|USS3275] [,TERM = terminal type] [,USSTAB = USS definition table name] [,VPACING = n|0|**2**]	
name	PU (switched SDLC line)	[ISTATUS = **ACTIVE**|INACTIVE] [,MAXLU = n|**2**] [,SPAN = (spanname)]	VSE VM
name	TERMINAL (BSC line)	ADDR = char ,TERM = 3275|3277|3284|3286 [,DLOGMOD = default logmode entry] [,FEATUR2 = ([**MODEL1**|MODEL2] [,PRINTR|NOPRINTR])] [,ISTATUS = **ACTIVE**|INACTIVE] [,LOGAPPL = application program name] [,LOGTAB = interpret table name] [,MODETAB = logon mode table name] [,SPAN = spanname] [,USSTAB = USS definition table name]	VSE VM
[name]	VBUILD	TYPE = ADJSSCP	
[name]	VBUILD	TYPE = APPL	
[name]	VBUILD	TYPE = CA [,CONFGDS = name] [,CONFGPW = password]	
[name]	VBUILD	TYPE = CDRM [,CONFGDS = name] [,CONFGPW = password]	
[name]	VBUILD	TYPE = CDRSC [,CONFGDS = name] [,CONFGPW = password]	

Name	Definition Statement	Operands	Restrictions
[name]	VBUILD	TYPE=DR	
[name]	VBUILD	TYPE=LOCAL [,CONFGDS=name] [,CONFGPW=password]	
[name]	VBUILD	TYPE=SWNET [,CONFGDS=name] [,CONFGPW=password] [,MAXGRP=n] [,MAXNO=n]	

Appendix B. Quick Reference for VTAM Start Options

<table>
<tr><th>Start Option</th><th>Restrictions</th></tr>
<tr><td>SSCPID=n</td><td></td></tr>
<tr><td>[,poolname=(baseno,bufsize,slowpt,F,xpanno,xpanpt)]</td><td></td></tr>
<tr><td>[,CDRSCTI=n|480]</td><td></td></tr>
<tr><td>[,COLD|WARM]</td><td>VM does not apply to V3R1.</td></tr>
<tr><td>[,CONFIG=xx|00|name]</td><td></td></tr>
<tr><td>[,CSALIMIT=0|n|nK|nM]</td><td></td></tr>
<tr><td>[,CSA24=0|n|nK|nM]</td><td>MVS/XA</td></tr>
<tr><td>[,DLRTCB=n|32]</td><td>MVS VM

VM the default in V3R1 is 8.</td></tr>
<tr><td>[,HOSTPU=host subarea PU name|ISTPUS]</td><td>VM does not apply to V3R1.</td></tr>
<tr><td>[,HOSTSA=n|1]</td><td></td></tr>
<tr><td>[,IOINT=n|180]</td><td></td></tr>
<tr><td>[,ITLIM=n|0]</td><td></td></tr>
<tr><td>[,LIST=yy]</td><td></td></tr>
<tr><td>[,MAXAPPL=n|10]</td><td>VM applies only to V3R1.</td></tr>
<tr><td>[,MAXSUBA=n|15]</td><td></td></tr>
<tr><td>[,MSGMOD=YES|NO]</td><td></td></tr>
<tr><td>[,NETID=network id]</td><td>MVS VM SNA network interconnection only.</td></tr>
<tr><td>[,NODELST=name]</td><td>VM does not apply to V3R1.</td></tr>
<tr><td>[,PPOLOG=YES|NO]</td><td>V3R1.1 only</td></tr>
<tr><td>[,PROMPT|NOPROMPT]</td><td></td></tr>
<tr><td>[,SONLIM=([m|60][,t|30])]</td><td></td></tr>
<tr><td>[,SSCPDYN=YES|NO]</td><td>VM does not apply to V3R1.</td></tr>
<tr><td>[,SSCPNAME=name]</td><td>MVS VM SNA network interconnection only.</td></tr>
</table>

<table>
<tr><th>Start Option</th><th>Restrictions</th></tr>
<tr><td>[,SSCPORD = <u>PRIORITY</u>|DEFINED]</td><td>VM does not apply to V3R1.</td></tr>
<tr><td>[,SUPP = <u>NOSUP</u>|INFO|WARN|NORM|SER]</td><td></td></tr>
<tr><td>[,TNSTAT[,CNSL|<u>NOCNSL</u>][,TIME = n|<u>60</u>]|<u>NOTNSTAT</u>]</td><td></td></tr>
<tr><td>[,TRACE|<u>NOTRACE</u>,ID = nodename,TYPE = BUF|IO[,EVERY]]</td><td></td></tr>
<tr><td>[,TRACE|<u>NOTRACE</u>,ID = linename,TYPE = LINE[,COUNT = n|<u>ALL</u>]]</td><td></td></tr>
<tr><td>[,TRACE|<u>NOTRACE</u>,ID = nodename,TYPE = SIT[,COUNT = n|<u>ALL</u>]]</td><td></td></tr>
<tr><td>[,TRACE|<u>NOTRACE</u>,ID = VTAMBUF,TYPE = SMS]</td><td></td></tr>
<tr><td>[,TRACE|<u>NOTRACE</u>,TYPE = VTAM]
[,MODE = <u>INT</u>|EXT]
[,OPTION = ALL|option|(option[,option...])]
[,SIZE = n|<u>2</u>]]</td><td>VM The default in V3R1 is TRACE TYPE = VTAM.</td></tr>
<tr><td>[,USSTAB = tablename]</td><td></td></tr>
<tr><td>[,VTAMEAS = n|<u>404</u>]</td><td>VM applies only to V3R1.</td></tr>
</table>

Appendix C. MVS TSO/VTAM System Programmer Information

TSO/VTAM provides the capability of using TSO through VTAM. TSO, formerly an option of the control program (hence the name time sharing option), is a standard feature in MVS that provides conversational time sharing.

TSO/VTAM supports the following terminals:

- IBM 3270 Information Display System
- IBM 3290 Information Display System
- IBM 3767 Communication Terminal
- IBM 3770 Data Communication System
- IBM 8775 Display Terminal
- IBM Displaywriter.

TSO/VTAM also can support, in conjunction with the IBM Network Terminal Option program product, the following non-SNA terminals:

- IBM 2741 Communication Terminal
- CPT-TWX Terminal
- World Trade Telegraph (WTTY).

The basic elements of TSO/VTAM are the terminal control address space (TCAS) and the VTAM terminal I/O coordinator (VTIOC). TCAS accepts logons from TSO/VTAM users and creates an address space for each user. VTIOC is the interface between TSO and VTAM; it coordinates data flow. TCAS and VTIOC, together with existing TSO components such as the LOGON scheduler, the application programs, the terminal monitor program, and the TSO service routines, make up a TSO/VTAM time-sharing system.

TSO/VTAM uses the VTAM application program interface (API). TSO/VTAM can reside with TSO/TCAM in the same host processor.

Information for system programmers about TSO/VTAM is found in the following sections:

- "VTAM Considerations" on page 326
- "MVS Considerations" on page 332
- "TSO/VTAM Considerations" on page 333
- "TSO Considerations" on page 337
- "How TSO/VTAM Differs from TSO through TCAM" on page 337.

VTAM Considerations

To use TSO/VTAM with VTAM, you must define the following to VTAM:

- TCAS and each TSO user
- TSO/VTAM session parameters.

For compatible logons, you must also create and define to VTAM an interpret table.

Defining TCAS and Each TSO User

As described in "Defining Application Program Major Nodes" on page 80, an APPL definition statement defines an application program to VTAM. Because TCAS and each TSO user are VTAM application programs, you must code APPL definition statements for them and put the statements in SYS1.VTAMLST.

Note: *To avoid obtaining unnecessary CSA storage for TSO users, you should define TSO applications to VTAM with EAS=1. Do not use the default (EAS=491).*

TSO/VTAM obtains CSA storage independently of CSALIMIT.

Single–Domain Network

In a single–domain network in which there are no overriding network naming conventions, you can use the following technique to code the APPL statements for TCAS and for the TSO users:

Code the following APPL statement for TCAS:

```
TSO      APPL    PRTCT=password,AUTH=(NOACQ,PASS,         C
                 NVPACE,TSO,NOPO),EAS=1
```

Code as many APPL statements, in the following format, as there will be users logged on to TSO/VTAM at one time:

```
TSOnnnn  APPL    PRTCT=password,AUTH=(NOACQ,PASS,         C
                 NVPACE,TSO,NOPO),EAS=1
```

If network naming conventions prevent the use of TSO and TSOnnnn as the name on the APPL statement, you may use any label name. In this case, you *must* use the ACBNAME operand to supply the ACBNAME for TCAS and for the TSO users as follows:

Code the following APPL statement for TCAS:

```
anytso   APPL    ACBNAME=TSO,PRTCT=password,                  C
                 AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

Code as many APPL statements, in the following format, as there will be users logged on to TSO/VTAM at one time:

```
anyuser  APPL   ACBNAME=TSOnnnn,PRTCT=password,          C
                AUTH=(NOACQ,PASS,NVPACE,TSO,NOPO),EAS=1
```

You must use the same *password* for TCAS and each TSO/VTAM user. A different application program name, in the form You must code TSO*nnnn,* for each session. *nnnn* is a decimal integer; the numbering must start with 0001 and must be sequential. Note that NOACQ and NOPO need not be coded; they are default values. See "The APPL Definition Statement" on page 82 for more information about application program names.

Multiple – Domain Network

For TSO sessions to be established in a cross – domain network, you must code CDRSC statements for each TSO/VTAM application program. Code the CDRSC statements in every VTAM host that owns logical units which will log on to the application.

In the VTAM host containing the TSO subsystem, it is not necessary to define the LUs in the other domains that will issue logons to TSO/VTAM if dynamic CDRSC definition is authorized. Also, the manager in the domain of the SLU may be authorized to create CDRSCs for the TSO application program associated with the TSO user. For information on coding CDRSCs, see "Defining Cross – Domain Resources" on page 208.

Code the following CDRM statements in each domain to avoid defining both the SLUs that log on to TSO/VTAM as CDRSCs in the TSO/VTAM domain, and the TSO user application programs as CDRSCs in the SLU's domain.

```
name1   CDRM   CDRDYN=YES,CDRSC=OPT,...

name2   CDRM   CDRDYN=YES,CDRSC=OPT,...
```

The CDRDYN operand determines whether a CDRM is authorized to dynamically create a CDRSC representing a cross – domain LU when a logon request is received from the LU (via a CDCINIT from the CDRM managing the LU).

The CDRSC operand determines whether the dynamic creation of CDRSC definitions is permissible when a logon request is received from this manager.

In a multiple – domain network where TSO/VTAM is to be run in more than one domain, you must have special definition statements with unique names for TCAS and the terminals in your domain and in interacting domains. Code the following APPL statement for TCAS in your domain:

```
tsoa        APPL   ACBNAME=TSO,PRTCT=password,         C
                   AUTH=(NOACQ,PASS,NVPACE,            C
                   TSO,NOPO),EAS=1
```

Code as many APPL statements in the following format as the maximum number of sessions that will be established with TSO/VTAM in your domain at one time:

```
tsoannnn     APPL     ACBNAME=TSOnnnn,PRTCT=password,      C
                      AUTH=(NOACQ,PASS,NVPACE,             C
                      TSO,NOPO),EAS=1
```

tsoa is a unique prefix that must be included on all APPL statements for your domain. The name, *tsoa*, must be unique in the network. You must use the same prefix (that is, *tsoa*) when coding APPL statements for each terminal logged on to the TSO/VTAM in this VTAM host concurrently. The *nnnn* suffix is a decimal integer that must start with 0001 and be sequential. The remaining operands are described above.

Code the following statement for each TCAS in another domain with which an LU in your domain will communicate:

tsob   CDRSC CDRM = name of VTAM manager for *tsob*

If dynamic cross – domain resource definition is not authorized, code the following CDRSC statements in each domain that contains an SLU that can communicate with your domain:

tsobnnnn   CDRSC CDRM = name of VTAM manager for *tsobnnnn*

and code a CDRSC statement in the domain of TSO/VTAM for each cross – domain SLU that can log on to TSO/VTAM.

tsob identifies the owning CDRM for each domain and must be included on each TCAS and CDRSC definition statement. The *nnnn* suffix is a decimal integer that must start with 0001 and be sequential.

Defining TSO/VTAM Session Parameters

VTAM needs to know the session protocols required by the terminals from which individual TSO users log on. This information is provided through logon mode tables, VTAM definition statements, and NCP generation definition statements.

Logon mode tables associate session protocols with device types. IBM provides a default logon mode table that describes session protocols for common devices and situations. However, you may need to replace or supplement the default table. *VTAM Customization* describes how to write your own logon mode tables and include them in VTAM. It also describes how to code VTAM definition statements and NCP generation definition statements to associate terminals with the appropriate logon mode tables.

In addition, a specific class of service name can be chosen. The class of service name to be used for the session is selected from the logon mode entry. If you omit the COS operand on the MODEENT macro instruction, the default COS is chosen. To choose a specific COS, you must include the COS name on the MODEENT macro instruction as follows:

MODEENT COS=cos name,logon mode parameters

3270 Characteristics

TSO/VTAM supports two types of 3270 devices: non-SNA devices attached locally or via a bisynchronous line protocol, and SNA devices attached by SDLC links. If no logmode table entry exists for a 3270 device, TSO/VTAM assumes non-SNA (LU0) and assumes the buffer size in the SCRSIZE parameter of the TSOKEY00 member of PARMLIB (either 480 [12 * 40] or 1920 [24 * 80]). For non-SNA devices with different screen sizes, or special features, and for SNA devices (LU2), you must provide a logmode table entry.

Use the MODETAB and MODEENT macro instructions to define 3270 characteristics in logmode tables and their entries. The PSERVIC parameter of the MODEENT macro instruction defines device LU type, buffer sizes, and QUERY capability (that is, programmed symbol sets, extended color, extended data stream, or extended highlight support).

Following are examples of definitions of MODEENT table entries. For further information on coding logon mode tables, see *VTAM Customization.*

The MODEENT Macro Instruction for Non-SNA 3270 Devices: Note that the FMPROF, TSPROF, PRIPROT, SECPROT, and COMPROT values shown below are the same as those used in the IBM-supplied logon mode table, ISTINCLM.

```
name      MODEENT    FMPROF=X'02',                 C
                     TSPROF=X'02',                 C
                     PRIPROT=X'71',                C
                     SECPROT=X'40',                C
                     COMPROT=X'2000',              C
                     PSERVIC=X'.....see below.....'
```

The MODEENT Macro Instruction for SNA 3270 Devices: The RUSIZES parameter below defines a 256-byte maximum secondary logical unit RU send size and a 1024-byte maximum primary logical unit RU send size. If the maximum primary logical unit RU size is 0, TSO/VTAM defaults to 1024 bytes.

```
name      MODEENT    FMPROF=X'03',                 C
                     TSPROF=X'03',                 C
                     PRIPROT=X'B1',                C
                     SECPROT=X'90',                C
                     COMPROT=X'3080',              C
                     RUSIZES=X'8587',              C
                     PSERVIC=X'.....see below.....'
```

The PSERVIC Operand of the MODEENT Macro Instruction: The PSERVIC operand of the MODEENT macro defines device LU type, buffer sizes, and QUERY capability (programmed symbol sets, extended color, or extended highlight support).

Code the 12 bytes of device-specific hexadecimal data of the PSERVIC operand as described below:

```
PSERVIC=
  X'00..00000000..........00' for non-SNA  (LU0)
  X'02..00000000..........00' for SNA  (LU2)

  X'..00....................' device without extended data
                              stream capability
  X'..80....................' device with extended data
                              stream capability

  X'............0000000001..' buffer size 480 only (12X40)
  X'............0000000002..' buffer size 1920 only (24X80)
  X'............0C280C507F..' buffer sizes 480 or 960
                              (12X40 or 12X80)
  X'............185020507F..' buffer sizes 1920 or 2560
                              (24X80 or 32X80)
  X'............18502B507F..' for buffer sizes 1920,3440
                              (24X80 or 43X80)
  X'............18501B847F..' for buffer sizes 1920,3564
                              (24X80 or 27X132)
  X'............18503EA07F..' buffer size 9920 only (62 * 160)
```

To prevent switching of screen sizes on a device that has more than one size possible, code the screen size you want in the primary area of PSERVIC and a X'7E', which means no switching. For example:

```
PSERVIC=X'............1B8400007E..' for 3564 buffer only
```

For details on coding the PSERVIC operand for your particular device, see your component description.

See *VTAM Programming* for details of the bit settings in the PSERVIC parameter, which represents bytes 13-24 of the session parameters.

The connection between a logmode entry and a device is made via VTAM operands on the LU, LOCAL, or TERMINAL definition statements during VTAM and/or NCP definition. Code the DLOGMOD operand to identify the name of a logmode table entry containing the session parameters to be used for the 3270 device being defined. Code the optional MODETAB operand to identify the logmode table containing the entry. If you do not code MODETAB, the IBM-supplied logmode table (ISTINCLM) is used.

You can also code the MODETAB and DLOGMOD operands on the GROUP, LINE, and CLUSTER definition statements. For details on planning the MODETAB, MODEENT, LU, LOCAL, and TERMINAL statements, see *Network Program Products Planning.* For information on coding the resource definitions and logon mode table macro instructions, see Chapter 5, "Defining the Network to VTAM" on page 69 and *VTAM Customization.*

Defining the 3790/3270 Configuration to TSO/VTAM

The 3270 attached to a 3790 (Version 7) uses the LU type 2 protocol. You must give session parameters to TSO/VTAM via a logmode table entry to properly identify the model number or screen size(s) of the device being used. The 3270 operator selects FPID 932 when logging on to the 3790 to activate the 3270 data stream compatibility function. During the logon procedure, the operator will be prompted for the application program ID and the logmode table entry containing the session parameters to be used for that session. The information is transmitted to VTAM by the Initiate Self request generated by the 3790. If the operator does not supply a logon mode table entry, the 3790 uses the standard logmode entry name EMU3790.

So the terminal operator will not have to know what logmode entry name to supply, create a separate logmode table for each 3270 model attached to the 3790 and include the default entry name EMU3790 in each table with the correct session parameters for that device.

Use the MODETAB and MODEENT macro instructions to define the logmode table and its entry. The PSERVIC operand on the MODEENT macro instruction carries the model number or screen sizes in row/column form. An example is given below for defining a logmode entry for a 3270 attached to a 3790 with a screen size of 1920 bytes (Model 2).

The MODETAB macro instruction:

```
LU2TABLE MODETAB
```

The MODEENT macro instruction:

```
EMU3790    MODEENT LOGMODE=EMU3790,                         C
                   FMPROF=X'03',                            C
                   TSPROF=X'03',                            C
                   PRIPROT=X'B1',                           C
                   SECPROT=X'90',                           C
                   COMPROT=X'3080',                         C
                   PSERVIC=X'020000000000000000000200'
```

See "The PSERVIC Operand of the MODEENT Macro Instruction" on page 330 for a description of the PSERVIC values.

The association between the logmode entry and the device is made on the LU definition statement during network definition to VTAM. Use the optional MODETAB parameter to identify the logmode table containing the entry. If MODETAB is not coded, the IBM−supplied logmode table (ISTINCLM) is used.

Defining 2741, TWX, or WTTY Terminals to TSO/VTAM

2741, TWX, or WTTY devices attached to a communication controller can be used with TSO/VTAM through its LU type 1 protocol managers. These devices are identified to VTAM during the NCP generation process. (See *Network Terminal Option Installation* and the *NCP−SSP Installation and Resource Definition Guide* for details.) Since the NCP will translate ASCII line code to EBCDIC for these devices, make sure that any logmode entry named by the DLOGMOD and MODETAB operands on the PU or LU definition statements identifies the device to VTAM as EBCDIC. Below is an example of an ASCII TWX logmode table entry definition that has the alternate code indicator in the COMPROT field set off (to indicate EBCDIC to TSO/VTAM).

The MODETAB macro instruction:

```
TWXTABLE MODETAB
```

The MODEENT macro instruction:

```
TWXDEVIC   MODEENT   LOGMODE=TWXDEVIC,                                C
                     FMPROF=X'03',                                    C
                     TSPROF=X'03',                                    C
                     PRIPROT=X'B1',                                   C
                     SECPROT=X'90',                                   C
                     COMPROT=X'3040'
```

Defining an Interpret Table for Compatible Logons

Because TSO/VTAM uses VTAM's logon facilities, the TSO LOGON command is not supported. This fact can be made transparent to terminal users if the system programmer defines an interpret table to allow logon requests to have the same format as that used by the TSO LOGON command. The following interpret table definition may be used:

```
INTBL     INTAB
          LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='LOGON'
          LOGCHAR APPLID=(APPLICID,TSO),SEQNCE='logon'
          ENDINTAB
          END
```

This interpret table allows a user to enter a logon command in uppercase or lowercase letters. The definition requires the characters *logon* to be coded in lowercase letters. This can be done by multipunching with a card punch.

MVS Considerations

You must also consider the MVS system requirements described below.

Defining TCAS Program Properties

TCAS must have an entry in the program properties table (PPT), load module IEFSD060 (CSECT IEFSDPPT). This makes certain designations as to execution attributes of TCAS:

- The program cannot be canceled.
- A unique protection key is assigned to the program.
- The program is privileged and is not swapped unless the address space is in a long wait.
- The program is a system task and is not timed.
- There is no host processor affinity.

The contents of the entry are an 8-byte EBCDIC field, IKTCAS00, followed by a 4-byte hexadecimal field, X'D860FFFF'.

After TSO/VTAM is installed, the PPT should be checked, because the complete PPT can be replaced during TSO/VTAM installation. The new PPT contains all current system entries. However, private user entries must be inserted into the new table.

Writing a Procedure for Starting TSO/VTAM Time Sharing

You must write a cataloged procedure for starting TSO/VTAM time sharing and include it in either SYS1.PROCLIB or your own procedure library.

Refer to: MVS or MVS/XA *SPL:TSO*

Creating a TSOKEY00 SYS1.PARMLIB Member

TSO/VTAM time sharing parameters determine how time sharing buffers should be controlled, the maximum number of users, and other options. Unless you want the IBM default parameters to apply, you must provide a SYS1.PARMLIB member containing the desired parameters. The parameters you may code, how to code them, and the IBM defaults are described in the MVS or MVS/XA *Initialization and Tuning* guide.

TSO/VTAM Considerations

Within TSO/VTAM itself, you should consider the following:

- Translation tables
- Coding TSO/VTAM exit routines
- Security
- Performance
- 3270 large screens.

Translation Tables

Translation tables allow TSO/VTAM users to replace internally those characters that are not available on a keyboard. If you call for character translation, translation tables—either your own or those supplied by IBM—are used.

Refer to: MVS or MVS/XA *SPL:TSO*

Coding TSO/VTAM Exit Routines

VTIOC and TCAS code contain hooks to exit routines that can be written by an installation to:

- Perform input and output editing that replaces or supplements IBM-supplied editing.
- Perform attention handling that replaces IBM-supplied attention handling.
- Provide support for terminals not supported by TSO/VTAM. (Note that an installation must write its own terminal input manager and terminal output manager, in addition to exit routines IKTGETXT, IKTINX2, and IKTINX1, to support terminals not supported by TSO/VTAM.)

The exit routines are optional. (If the user does not write the exit routines, an unresolved external reference message will be received.) Before an exit routine can be used it must be link-edited with the object module that calls it. At appropriate points during VTIOC and TCAS processing, a check is made to determine if a particular exit routine exists. If it exists, it is called; if it does not exist, normal processing continues.

The exit routines are summarized in Figure 40. For more information about exit routines, see *VTAM Customization.* For information about coding your own exit routines, see *VTAM Programming.*

Name	Purpose	Terminal Type	Caller
VTIOC			
IKTGETXT	Edit input data	non-supported	IKTVTGET
IKTIDSX1	Replace or supplement IBM-supplied output editing	3270	IKT3270O
IKTIDSX2	Supplement IBM-supplied input editing	3270	IKT3270I
IKTIDSX3	Supplement IBM-supplied attention handling	3270 (LU0)	IKT3270I
IKTIDSX4	Replace or supplement IBM-supplied input editing	3270	IKTVTGET
IKTINX2	Initialize user-written I/O managers	non-supported	IKTXINIT

Figure 40 (Part 1 of 2). TSO/VTAM Exit Routines

Name	Purpose	Terminal Type	Caller
IKTRTX1	Replace or supplement IBM – supplied output editing	3767/3770 2741	IKT3767O
IKTRTX2	Supplement IBM – supplied attention handling	3767/3770 2741 WTTY TWX	IKT3767I
IKTRTX3	Replace IBM – supplied attention handling	3767/3770 (LU1)	IKTMLU1
IKTRTX4	Replace or supplement IBM – supplied input editing	3767/3770 (LU1)	IKTVTGET
IKTWTX1	Replace or supplement IBM – supplied output editing	TWX WTTY	IKTWTTYO
TCAS			
IKTCASX1	Replace or supplement IBM – supplied logon error messages	non – supported	IKTCAS31
IKTINX1	Set terminal type and buffer size	non – supported	IKTCAS23

Figure 40 (Part 2 of 2). TSO/VTAM Exit Routines

Security

The TSO subsystem is considered a *secure* application. That is, confidential data is handled on behalf of the user in ways that prevent unauthorized disclosure of the data. The CONFTXT parameter in the TSOKEY00 member of PARMLIB determines whether output data will be considered confidential text. The default is that the data will be considered confidential. TSO/VTAM protects tracing of user data by setting the CONFTXT indicator in the NIB at the time the user logs on.

If CONFTXT = NO, VTAM can perform buffer or IO traces on the data. If CONFTXT = YES (the default), the data is considered confidential and no data is recorded. The CONFTXT parameter, however, does not apply to the TSO type VTAM trace for TPUT/TPG/TGET buffers; these are always traceable. See *Initialization and Tuning* for details.

Performance

The following list contains suggestions for improving TSO/VTAM performance:

- Terminal users should stack input data whenever possible. This means using the multiple–line input technique on 3270 terminals and buffered SDLC transmission on 3767 and 3770 terminals. Stacking input results in a decreased number of data transmissions to TSO. Users of 3270 terminals can stack up to a full screen of data; 3767 and 3770 terminal users can stack from 256 bytes to 2,000 bytes of data, depending on the terminal's buffer size.

- Users of 3767 and 3770 terminals can reduce idle time by "typing ahead." This means that whenever the terminal is not receiving data or is transmitting data, the user can enter input.

- Use BREAK mode with 3276 and 3278 terminals to expedite unlocking the keyboard. BREAK mode reduces idle time by allowing typing ahead.

- Use automatic line numbering on 3270 terminals to shorten the TSO/VTAM processing path when handling input; this reduces system overhead and expedites unlocking the keyboard.

- Avoid use of autologon since looping will occur if the terminal is powered off.

3270 Large Screen Considerations

The following information describes screen management techniques for IBM 3270 terminals that have default and alternate screen sizes determined by the logon mode table entries used for them at logon. The default size is accessed by sending an Erase–Write (EW) command to the terminal, and the alternate size is accessed by sending an Erase–Write–Alternate (EWA) command.

TSO/VTAM Screen Management

TSO/VTAM can manage the screen for TSO application programs that use line–oriented I/O. In this case, TSO/VTAM uses the maximum screen size for the device, whether it is the default or the alternate in the logon mode table entry. When screen erasure is necessary, TSO/VTAM uses either the EW or the EWA command, as required to access the larger screen size.

Full–Screen Application Program Screen Management

An IBM–supplied or user–supplied application program that uses full–screen I/O in TSO full–screen mode can be used to manage the screen (see MVS or MVS/XA *TSO Guide to Writing a Terminal Monitor Program*). TSO/VTAM uses the screen size used by the application program. If the CLEAR key is pressed, TSO/VTAM erases the screen with either an EW or

EWA command depending on which was last used by the application program.

TSO Considerations

Information about the TSO-related steps that must be performed before a TSO/VTAM terminal user logs on can be found in the MVS or MVS/XA *SPL:TSO* and *Initialization and Tuning* guide. (Make sure you have updated these manuals with TSO/VTAM supplements before referring to them.) The following list outlines the steps and refers the reader to the appropriate publication:

- Write LOGON cataloged procedures and include them in SYS1.PROCLIB. Refer to MVS or MVS/XA *SPL:TSO.*
- Construct the TSOKEY00 member or an alternate member of SYS1.PARMLIB (or an alternate data set) to set VTIOC parameters. Refer to the MVS or MVS/XA *Initialization and Tuning* guide.
- Include SYS1.CMDLIB in a LNKLSTxx member of SYS1.PARMLIB or in a LOGON cataloged procedure. Refer to the MVS or MVS/XA *Initialization and Tuning* guide.
- Create or convert the user attribute data set (UADS) and the broadcast data set. Refer to the MVS or MVS/XA *SPL:TSO* manual.
- Build translation tables if character translation is desired. Refer to the MVS or MVS/XA *SPL:TSO* manual.
- Write the procedure that starts TSO/VTAM time sharing. Refer to the MVS or MVS/XA *SPL:TSO* manual.
- Write any command exit routines you plan to use. Refer to the MVS or MVS/XA *SPL:TSO* manual.

How TSO/VTAM Differs from TSO through TCAM

The following list describes differences between TSO/VTAM and TSO through TCAM. It is intended for system programmers who are familiar with TSO through TCAM. The differences are:

- TSO/VTAM does not use the TCAM message control program (MCP).
- VTIOC replaces TIOC (terminal I/O coordinator) as the means for controlling data movement between TSO and the access method.
- TSOKEY00 replaces IKJPRM00 as the SYS1.PARMLIB member that contains time-sharing parameters.
- The system START command starts TSO/VTAM time sharing. The MODIFY command controls the number of users that may be logged on

and terminates user address space. The STOP command stops TSO/VTAM time sharing.

- TSO/VTAM user address spaces are totally independent of each other. Each address space is an individual TSO/VTAM application program, with ownership of its terminal. A noncontiguous terminal status block (TSB) is allocated dynamically when each user logs on. Users do not share buffers; VTIOC allocates buffers as needed.

- TSO/VTAM uses VTAM's logon facilities. The existing TSO LOGON command, which provides a logon to the TSO terminal monitor program (TMP), is supported only for re-logons. The existing TSO LOGOFF command is supported only if HOLD is not coded.

- TSO/VTAM support for the PROFILE and TERMINAL commands differs from that provided through TCAM. You can define character-delete and line-delete control characters (PROFILE command) only for the LU type 1 devices supported by TSO/VTAM, because of editing capabilities of TSO/VTAM-supported terminals. (Refer to the list of supported devices in *Network Program Products General Information.*) Definition of attention interruption characteristics (TERMINAL command), including the simulated attention interruption, is not supported, because attention interruptions are always accepted on TSO/VTAM-supported terminals. Character translation (TERMINAL command) is provided for TSO/VTAM-supported terminals.

- There are some considerations for terminal control macro instructions. STTRAN sets character translation. STFSMODE turns 3270 full-screen mode on or off. STLINENO is the number of a 3270 screen line on which the next non-full-screen message should appear and optionally turns full-screen mode on or off. STTMPMD determines whether the terminal control routine is active for the terminal and whether the ATTN and CLEAR keys are to be passed to the application program as data. GTTERM obtains the primary or alternate screen sizes and places them in a user-provided area. The STATTN macro instruction is not supported by TSO/VTAM. The STCC macro instruction is supported for NTO-attached devices.

- No data is lost when a user logs on again (reconnects) after his line was disconnected.

- TSO/VTAM takes advantage of multiprocessing because it is multitask structured; TCAM executes in one host processor at a time.

Appendix D. MVS Filing Cryptographic Keys

If you use the VTAM data encryption facility, you must file cryptographic keys on the cryptographic key data set at each host processor before activating any LUs that are to be used in cryptographic sessions.

Filing Secondary Logical Unit Keys for Single–Domain Cryptographic Sessions

To allow cryptographic sessions to be established within a single domain:

- Install the IBM Programmed Cryptographic Facility. Refer to *IBM Programmed Cryptographic Facility Installation Reference Manual.*

- Use the IBM Programmed Cryptographic Facility to file secondary logical unit (SLU) keys on the cryptographic key data set (CKDS) as follows.

  For each device–type LU that is to be used as the secondary end of a cryptographic session code:

  LOCAL *name*

  Where *name* is the name of the LU. This LOCAL statement generates an SLU key for the LU and adds it to the CKDS enciphered under the first variant of the host master key. It also returns a clear SLU key. Enter the clear SLU key into the device.

  For each VTAM application program that is to be the secondary end of a cryptographic session, code:

      REMOTE *name*

  Where *name* is the name of the application program. This REMOTE statement generates an SLU key for the application program and adds it to the CKDS enciphered under the second variant of the host master key.

Filing CDRM Keys for Cross-Domain Cryptographic Sessions

To allow cross-domain cryptographic sessions to be established, file SLU keys for each domain as described above. Then use the IBM Programmed Cryptographic Facility to file cross-domain keys on the cryptographic key data set (CKDS) at each host processor as follows.

- For each pair of host processors (HOST1 and HOST2) that are to have cross-domain cryptographic sessions between their domains, at HOST1 code:

  CROSS *name*

  Where *name* is the name of HOST2's CDRM. This CROSS statement generates two cross-domain keys, one defined as local and the other defined as remote. It stores the local cross-domain key in the CKDS enciphered under the first variant of HOST1's host master key and stores the remote cross-domain key in the CKDS enciphered under the second variant of HOST1's host master key. Both of these keys are associated with the name of HOST2's CDRM. This CROSS statement also returns clear copies of the two keys.

- The cross-domain keys generated at HOST1 must be taken to HOST2 and supplied as input to the IBM Programmed Cryptographic Facility in a CROSS statement as follows:

  CROSS name,KEYLOC=x,KEYREM=y,ADD

  Where:

  - *name* is the name of HOST1's CDRM.
  - *x* is the clear remote cross-domain key from HOST1.
  - *y* is the clear local cross-domain key from HOST1.
  - This CROSS statement adds the two cross-domain keys to HOST2's CKDS, associates them with the name of HOST1's CDRM, reverses the local/remote relationship of the two keys, and enciphers key *x* and key *y* under the first and second variants of HOST2's host master key respectively.

  Note: The terms local and remote in reference to the keys used by the CROSS statement do not mean the same as the same terms used in other contexts in this manual. Refer to:

  - *OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility Installation Reference Manual*
  - *OS/VS1 and OS/VS2 MVS Programmed Cryptographic Facility General Information Manual.*

Appendix E. VM VM SNA Console Support (VSCS)

VM/SNA Console Support (VSCS) is an application program that acts as an intermediary between VTAM and a VM system. If an end user in a VTAM network wants to log on to VM, he must first log on to the VSCS application that is running in the VM host. VSCS, however, can be made completely transparent to end users. End users need not be aware of VSCS unless you want them to be.

See *VTAM Operation* for a detailed description of LOGON and LOGOFF procedures in a system that has VSCS.

See *Network Program Products General Information* for a list of the devices supported by VSCS as virtual machine consoles and a list of the output devices supported by VSCS.

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.

See Figure 12 on page 36 for an example of a VM directory entry that is coded so that VSCS can use CCS and IUCV.

Note that if VSCS runs in a separate virtual machine from VTAM, it must be in an authorized machine.

The DTIGEN Macro Instruction

VSCS has its own set of start options. You code them on a DTIGEN macro and place the macro in a CMS file called DTIUSERn ASSEMBLE (where *n* is a number between 0 and 9) on the VTM191 disk. Assemble this file using VTAMAC MACLIB.

DTIGEN is an assembler macro instruction, so code it in a format like this. (The value *5* is only an example.)

```
[symbol] DTIGEN DTIUSER=5,PRTSHR=N,CSTRACE=N,...
         END
```

The operands are “stacked” in the following description to make them easier to read, but in practice you can code them on one line. (You can put a non-blank character in column 72 if the list of operands continues onto another line. The continuation line must begin in column 16.)

All the operands on the DTIGEN macro are optional. They are divided into categories according to their purpose.

The format of the DTIGEN macro is:

Name	Macro	Operands
[symbol]	DTIGEN	[IDENT=identifier|**yyddd**] [,DTIUSER=n|**0**] [,APPLID=name|**VM**] [,PASSWRD=password] [,DPXMTL=n|**1724**] [,KPXMTL=n|**256**] [,PRNTNUM=n|**16**] [,RCVBFRL=n|**256**] [,RPLNUM=n|**8**] [,PRTSHR=**Y**|N] [,TIMECPY=n|**3**] [,TIMEREL=n|**120**] [,BFRFIFO=Y|**N**] [,CSTRACE=**Y**|N] [,DPTRACE=Y|**N**] [,VTTRACE=Y|**N**] [,TRASIZE=n|**250**] [,LGNCMDS=pre-logon command table] [,RDSPTMR=n|**10**] [,W3767L=n|**129**] [,W2741L=n|**129**] [,WTWXL=n|**72**] [,VSAMLM=n|**10**] [,DPACE=n|**screen depth**] [,KPACE=n|**10**] [,LURTRY=n|**5**] [,TSKRTRY=n|**10**] [,ACBLOOP=n|**3**] [,DUMP=Y|**N**] [,BLKMULT=n|**1**] [,DEXIT=Y|**N**] [,KEXIT=Y|**N**]

symbol
is optional. If you assign a symbol, it must be from 1 to 8 alphameric characters, beginning with a character other than the dollar sign.

IDENT=identifier|yyddd
is a 1- to 5-character EBCDIC identifier. It is used only for debugging—it identifies parameters being used by VSCS when a dump is taken. The default is the Julian date (yyddd) on which the DTIGEN macro is assembled.

DTIUSER=n|<u>0</u>

determines the name of the ASSEMBLE file that contains this DTIGEN macro. Its value, *n*, should appear in the filename: DTIUSER*n* ASSEMBLE. *n* can be any number from 0 to 9.

Using this operand, you can maintain more than one module, each containing a different set of startup parameters; each time you start VSCS you can select the module that contains the parameters you need.

VTAM – Related Operands

APPLID=name|<u>VM</u>

is the VTAM application program identifier. VSCS uses this operand for the VTAM OPEN ACB macro. It is also used as the IUCV application program identifier. *name* can be 1 to 8 alphanumeric characters. The first character must be alphabetic. This value must match the name given for VSCS on the VTAM APPL statement. The default is VM. (See "Defining the VSCS Application to VTAM" on page 350 for more information.)

You can also customize this name with an interpret table or a USS table. See *VTAM Customization* for details.

PASSWRD=password

identifies the VTAM application password in 1 to 8 EBCDIC characters. If you code a PRTCT operand on the APPL definition statement for VSCS, this password must match it. ("Defining the VSCS Application to VTAM" on page 350 describes how to code the APPL statement.) If you omit PRTCT on the APPL statement, VTAM does no password checking.

The default is the value you coded for APPLID.

DPXMTL=n|<u>1724</u>

is 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, this value should be the average size of the data stream transmitted to a display terminal operating in line mode. This is usually the average number of lines of output sent to the terminal before a RECEIVE or READ is requested.

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 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 *VTAM Customization* for the format of the VSCS accounting record.)

The range is 256 to 32600. The default is 1724. This is the size of a 24 * 80 screen, minus the input area and the status area.

KPXMTL = n|256

is the buffer size, in bytes, for data being sent to keyboard terminals and 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.

The range is 138 to 32600. The default is 256 bytes.

PRNTNUM = n|16

is the maximum number of printers allowed to be logged on for PF – key copy requests. Choose a number that will accommodate a normal increase in the number of printers in the network; since PRNTNUM requires 32 bytes for each printer, however, choosing a number that is unnecessarily high wastes storage.

The range is 1 to 32600 The default is 16.

RCVBFRL = n|256

is the buffer length, in bytes, for VTAM RECEIVE requests issued by VSCS. In general, this 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. (Refer to *VTAM Customization* for the format of the VSCS accounting record.)

The range is 80 to 32600. The default is 256 bytes.

RPLNUM = n|8

is 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 may 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 range is 1 to 16; the default, 8, is the recommended value. However, if you are running more than 40 terminals the maximum value of 16 is recommended.

PRTSHR = Y|N

Y enables VSCS dynamic printer reallocation and sharing. If the VSCS RELREQ exit is driven, and the time designated by TIMEREL has elapsed since its last use, it immediately releases the associated printer. Otherwise, VSCS waits until the interval has elapsed before it releases the printer. Also, when VSCS receives a PF – key copy request for a printer that was previously known to it, it will attempt to reacquire the printer. The default is Y.

TIMECPY = n|3

defines the printer allocation time limit in seconds. *n* is the maximum time that VSCS waits to reacquire a printer it previously released, before causing a PF – key copy request to fail. A PF – key copy request is rejected if the time required to allocate a printer exceeds this limit.

The range is 0 to 30 seconds. The default is 3 seconds. This operand is ignored if PRTSHR = N.

TIMEREL = n|120

defines the printer deallocation time limit in seconds. *n* is the minimum time that VSCS will keep a printer after its last use. VSCS will not release a printer to another application until this interval has elapsed.

The range is 0 to 86400 seconds (24 hours). The default is 120 seconds (2 minutes). This operand is ignored if PRTSHR = N.

Trace – Related Operands

Note: For more information on the traces mentioned below, see the *VTAM Diagnosis Guide.*

BFRFIFO = Y|N

tells whether dynamic buffers are to be selected from the dynamic buffer storage queues on a FIFO (first – in first – out) basis. The default is N, meaning that the buffers are selected on a LIFO (last – in first – out) basis.

See the descriptions of the trace operands below for additional information. Also see the description of the BLKMULT operand on page 349 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. Using this option may affect system performance, and it should normally be initiated by an operator command when necessary (see the description of VSCS trace commands below).

CSTRACE=Y|N

CSTRACE=N turns off the Console Communications Services (CCS) trace. This option traces the data flow between VSCS and CCS, within VSCS. There is another trace facility in CP to trace data flow within CCS. The default is Y, which leaves the CCS trace on.

Note: Using this option may affect system performance, and it should normally be initiated by an operator command (VSCS TRACEON CCS) when necessary.

DPTRACE=Y|N

DPTRACE=Y turns on the VSCS dispatcher trace. This option records information about VSCS before it dispatches control to the components that process the various tasks. The default is N, which leaves the VSCS dispatcher trace turned off.

Note: Using this option may affect system performance, and it should normally be initiated by an operator command (VSCS TRACEON DISP) when necessary.

VTTRACE=Y|N

VTTRACE=Y turns on the VTAM trace. This option traces the data flow between VSCS and VTAM at the application program interface (API). The default is N, which leaves the VTAM trace turned off.

Note: Using this option may affect system performance, and it should normally be initiated by an operator command (VSCS TRACEON VTAM) when necessary.

TRASIZE=n|250

is 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, and VTTRACE) that are active concurrently. (Alternatively, the operator, when issuing the VSCS TRACEON command, can request that trace data be recorded on an external medium.)

The trace entries are preceded by a 64−byte trace table header that contains table control information. 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:

(TRASIZE + 3) * 32

This storage is allocated when the first trace entry is made and is freed only when VSCS is terminated.

The range is 125 to 32600 entries. The default is 250 entries.

CCS – Related Operands

LGNCMDS = pre – logon command table

lists 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 simply 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. Thus 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 table below, 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,*)
```

RDSPTMR = n|<u>10</u>

is the redisplay timer value, in tenths of a second. It sets the time that elapses before input is redisplayed on the screen. This is done to reduce the number of writes to the terminal. If output comes back during this time, it is displayed immediately; otherwise the input is redisplayed and then the output is displayed when it arrives.

The range is 0 to 255. The default is 10 (1 second), which is the recommended value. If you set the value higher than about 30 (3 seconds), users may believe their terminals are hung, or complain of slow response time, when they do complex operations.

W3767L = n|<u>129</u>

is the default line size for 3767 and 3777 terminals. This is the initial setting of the terminal output line length for a 3767 or a 3777. The user can change this by issuing the CP TERMINAL LINESIZE command and supplying a new line length.

The range is 1 to 255 characters. The default is 129.

W2741L = n|<u>129</u>

is the default line size for 2741 terminals. This is the initial setting of the terminal output line length for a 2741. The user can change this by issuing the CP TERMINAL LINESIZE command and supplying a new line length.

The range is 1 to 255 characters. The default is 129.

WTWXL = n|<u>72</u>

is the line length for TWX keyboard printers. This is the initial setting of the terminal output line length for a TWX device. The user can change this by issuing the CP TERMINAL LINESIZE command and supplying a new line length.

The range is 1 to 255 characters. The default is 72.

VSAMLM = n|10

is the maximum number of concurrent messages that can be transferred between VSCS and CCS using IUCV. VSCS transfers data to and from CCS using IUCV CONNECT and IUCV ACCEPT. The value of this operand defines the largest number of concurrent paths between VSCS and CCS for sending and receiving messages.

The range is 4 to 255. The default is 10.

DPACE = n|screen depth

is the pacing value between CCS and VSCS to be used with display terminals. The value of this operand determines how many messages CCS will send to VSCS before waiting for a response.

The range is 1 to 255. The default is the number of rows on the display terminal (the screen depth), and is device-dependent. It is strongly recommended that you use the default.

Note: This pacing option is not related to the PACING and VPACING options available in VTAM and NCP.

KPACE = n|10

is the pacing value between CCS and VSCS to be used with keyboard terminals and printers. The value of this operand determines how many console mode messages CCS will send to VSCS before waiting for a response. Note that CP may not respond to any of the attention keys until this number of messages has been sent to the device.

The range is 1 to 255. The default is 10. It is strongly recommended that you use the default.

Note: This pacing option is not related to the PACING and VPACING options available in VTAM and NCP.

Recovery-Related Operands

LURTRY = n|5

is the logical-unit retry count. If a specific LU causes repeated ABENDs (abnormal ends), this retry count tells how many times it should be reactivated. When the count is exceeded, VSCS disconnects the LU from VM and then releases it back to the SNA network. This number should be less than the TSKRTRY count so that one user cannot cause VSCS to terminate.

The range is 1 to the value of TSKRTRY. The default is 5 or the TSKRTRY value, whichever is smaller.

TSKRTRY = n|10

is the task retry count (maximum number of retries of a task that ABENDs) before VSCS termination is invoked. Task retry occurs only for data being sent to a logical unit after VSCS initiates logical unit purge for the affected logical unit. This number should be greater than the LURTRY count.

The range is 1 to 255. The default is 10.

ACBLOOP=n|3
is the amount of time, in minutes, that VSCS will retry opening its VTAM ACB for certain error conditions. 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.

The range is 0 (no retry) to 32767. The default is 3.

DUMP=Y|N
Y causes VSCS to take dumps in certain internal error situations. The default is N: VSCS will not take dumps.

Note: VSCS always takes dumps when an ABEND occurs, regardless of how you set this operand.

Storage–Related Operand

BLKMULT=n|1
is the decimal number used as a multiplier in determining the number of blocks in a segment.

VSCS maintains a number of fixed–length storage pools. VSCS allocates an area from a pool that has blocks equal to or greater than the size of the request. Storage is obtained in amounts called segments, which are divided into fixed–length blocks.

The range is 1 to 4. The default, and the recommended value, is 1. A higher value results in fewer storage requests but requires more storage. For example, if BLKMULT=4, then VSCS obtains four times as much storage for each request; each request for 4K would obtain 16K of storage.

Operands to Activate User Exits (V3R1.1 Only)

DEXIT=Y|N
tells whether exit routines which translate data being sent two and from display terminals, are to be active. VTAM provides two such user exits: DTIPDDSO for data sent to the terminal, and DTIPDDSI for data received from the terminal. If you wish to code your own user exits for data translation, they must have these same names and must follow the rules listed in "Coding Exit Routines for Translating Data" on page 352.

The default is N, which means that the user exits are not activated.

KEXIT=Y|N

tells whether exit routines which translate data being sent two and from keyboard terminals and printers, are to be active. VTAM provides two such user exits: DTIPDNTO for data sent to the device, and DTIPDNTI for data received from the device. If you wish to code your own user exits for data translation, they must have these same names and must follow the rules listed in "Coding Exit Routines for Translating Data" on page 352.

The default is N, which means that the user exits are not activated.

VTAM Considerations

When you install VTAM, you must take some additional steps before you can run VSCS. "Installing VTAM" on page 39 explains the process of installing VTAM and VSCS in a virtual machine in a GCS group. Besides the activities described in that section, you must define the VSCS application to VTAM. You may also want to:

- Code logon mode tables for VSCS devices
- Code USS tables to tailor the USS LOGON and LOGOFF commands to the needs of your installation.

Note: When you first install VTAM and VSCS, there are default logon mode tables and USS LOGON and LOGOFF commands for VSCS devices. For most uses they will meet your needs.

The following sections explain what you have to do to define the VSCS application to VTAM, code logon mode tables, and code USS tables.

Defining the VSCS Application to VTAM

VSCS is a VTAM application program and must be defined to VTAM as an application program major node. As described in "Defining Application Program Major Nodes" on page 80, an application program major node is defined by filing one VBUILD statement for the major node and an APPL statement for each application in the major node. ("Filing Network Definition Statements" on page 70 explains how to file network definition statements.)

The following example shows the definition statements required to define the VSCS application to VTAM:

```
ISTAPPLS VBUILD TYPE=APPL
VM       APPL   AUTH=(PASS,ACQ),ACBNAME=VM,PRTCT=VM,AUTHEXIT=YES,
                SONSCIP=YES
```

Notes:

1. *ISTAPPLS VTAMLST is the name of the file containing the definition statements for the VSCS application major node. You may use a different name if you choose.*

2. *If you code an ACBNAME, it must match the name you use on the APPLID operand of all DTIGEN macros you code. (See "The DTIGEN Macro Instruction" on page 341 for more information on DTIGEN.) If you do not code an ACBNAME, the default name is VM.*

3. *AUTHEXIT=YES is required.*

4. *Do not code any operands on the APPL statement other than the ones shown in the example.*

Coding Logon Mode Tables for VSCS Devices

Logon mode tables are necessary to define session parameters for devices supported by VSCS. The default logon mode table is named ISTINCLM.

See *VTAM Customization* for additional information about coding logon mode tables in VTAM.

Coding USS Tables for Logging On and Off

You may code USS tables to tailor the USS LOGON and LOGOFF commands to the needs of your terminal operators. When you first install VTAM and VSCS, there are default USS mode tables for the USS LOGON and LOGOFF commands. They are described in *VTAM Customization.*

VTAM Customization contains sample USS tables that you can code, should you wish to code your own USS table for a VSCS terminal.

Defining Printers

You can define a list of printers to VSCS without establishing sessions between VSCS and the printers. This is useful when users on different VM systems share the same printer for PF-key copy functions. You do this by including a VSCS command, PRINTER, in the PROFILE GCS for the VTAM and VSCS virtual machine. This PROFILE GCS is described in "The PROFILE GCS for the VTAM and VSCS Virtual Machine" on page 48.

The syntax of the PRINTER command is:

PRINTER printer-name printer-name printer-name. . .

For details, refer to *VTAM Operation.*

Notes:

1. *Once a printer is defined to VSCS with the PRINTER command, you cannot delete the definition without restarting VSCS.*

2. *The number of printers you define cannot exceed the value of the PRNTNUM operand on the DTIGEN macro.*

3. *If you want these printers to become active as soon as VTAM is started, include them in an ATCCONxx VTAMLST file. "Creating Start Option and Configuration Lists" on page 221 contains a detailed discussion of ATCCONxx VTAMLST files.*

Coding Exit Routines for Translating Data

You can code exit routines to modify a data stream before VSCS translates it or to bypass VSCS translation altogether. VTAM provides four exit routines: DTIPDDSO for data being sent to a display device, DTIPDDSI for data being received from a display device, DTIPDNTO for data being sent to a keyboard device, and DTIPDNTI for data being received from a keyboard device. They do nothing except set a return code of 0, which allows VSCS to translate the data.

All such exit routines must conform to the following rules:

- The exit routine cannot change the size of the data stream. (For this reason, APL and TEXT data translation are not allowed.)

- A return code of 0 allows VSCS to translate the data; a return code of 8 prevents VSCS from translating the data.

- On exiting, register 13 must have the same contents—the address of a save area—as it had on entry.

- The exit routine is responsible for handling any responses from CP or from the logical unit as a result of the data change.

Offset	Description	
+00	Address of data	
+04	Length of data stream	
+08	Device attributes:	
	X'80'	Alternate size available
	X'40'	New APL/TEXT (3278 or 3279)
	X'1C'	Color/highlight functions:
	X'10'	Extended data stream available
	X'08'	Extended color available
	X'04'	Extended highlighting available
+09	Character set being used:	
	X'00'	EBCDIC
	X'01'	APL
	X'02'	TEXT
+0A	State of the device:	
	X'00'	No MORE or HOLDING condition
	X'01'	No MORE or HOLDING condition (CMS edit mode)
	X'02' – X'03'	MORE condition, 50 second timer set
	X'04' – X'05'	MORE condition, 10 second timer set
	X'06' – X'07'	HOLDING condition
	X'08' – X'0F'	NOT ACCEPTED logic applies
	X'10' – X'1F'	PF key copy in progress
	X'20'	Full-screen support mode
	X'40'	Full-screen support mode
	X'60'	Full-screen support mode
	Note: X'80' through X'88' are VSCS internal modes.	
	X'80'	Initial logon in progress
	X'81'	Logon on device
	X'82'	CP connect in progress
	X'83'	Logging on, awaiting initial console request
	X'84'	LU severed from CP, but in session with VSCS
	X'85'	Logoff in progress, awaiting CPBRK
	X'86'	Conditional logoff in progress, awaiting final send completion
	X'87'	Sever required at message complete
	X'88'	LU purged; VSBRK scheduled to VTAM services
+0B	Reserved	
+0C	Screen width	
+0E	Screen depth	

Note: The screen width and depth give the physical size of the screen.

Figure 41. Register 1 Contents Passed to Exit Routines (for Display Devices)

Exit routines for **display devices** receive control with the following information:

Register 1	Parameter list (see Figure 41 on page 353)
Register 13	Save area for the caller
Register 14	Return address
Register 15	Entry point of the exit routine.

Exit routines for **keyboard devices** receive control with the following information:

Register 1 Parameter list:

+00	Address of data
+04	Length of data stream
+08	Model of terminal or printer (01 through 05)

Register 13	Save area, in DTISWB, for use by the exit routine
Register 14	Return address
Register 15	Entry point of the exit routine.

You can activate these exit routines by setting the operands DEXIT (for display devices) and KEXIT (for keyboard devices) on the DTIGEN macro. See “Operands to Activate User Exits (V3R1.1 Only)” on page 349 for details.

Defining Your Own Set of VSCS Start Options

You should code your own set of start options for VSCS if the default start options do not meet your needs. You may code up to ten different sets of start options for VSCS. Each set must be in a unique DTIUSERn ASSEMBLE file: each file must contain exactly one DTIGEN macro, and each file must be assembled separately, using VTAMAC MACLIB. Link-edit resulting text files into VSCSUSER LOADLIB on the VTM191 minidisk using the CMS VMFLKED command.

When you code the DTIGEN macro, code values only for those start options you wish to change. The default value will be assumed for each start option that you do not code.

Follow these steps to code, assemble, and link-edit a set of start options for VSCS. Label the set of start options DTIUSERn, where the value of *n* depends on what you code for the DTIUSER operand; see the description of the DTIUSER operand on page 342. (*n* is also the value you assign to the PARM operand of the VSCS START command when you start VSCS.) In this example, the set of start options is labeled DTIUSER5.

1. Access the VTM191 disk as the A-disk.

2. Create a CMS file named DTIUSER5 ASSEMBLE on the A-disk. In this file, code one DTIGEN macro instruction with operands for the start options that you wish to change. See "The DTIGEN Macro Instruction" on page 341 for the format of the DTIGEN macro.

3. Identify the macro library to the assembler by issuing the command GLOBAL MACLIB VTAMAC.

4. Issue the CMS ASSEMBLE command to assemble the DTIUSER5 ASSEMBLE source code. The output of the assembler is a file named DTIUSER5 TEXT A.

5. Code a link-edit script in a CMS file named VSCSUSER LKEDCTRL on the A-disk. The link-edit script must contain the following statements:

   ```
   %LIBRARY VSCSUSER
    INCLUDE DTIUSER5
    NAME DTIUSER5(R)
   ```

   The purpose of the statements is:

command	**purpose**
INCLUDE	Tells VMFLKED EXEC to use the file named DTIUSER5 TEXT as input to the linkage editor.
NAME	Tells the linkage editor to name the output load module DTIUSER5.

6. Link-edit the DTIUSER5 TEXT deck by issuing the command:

   VMFLKED VSCSUSER

   This command uses the VSCSUSER LKEDCTRL file to build the DTIUSER5 load module in VSCSUSER LOADLIB on the VTM191 disk.

7. In the EXEC used to start VSCS (an example appears in Figure 15 on page 48), insert the name VSCSUSER after the word LOADLIB on the GLOBAL command. For example the GLOBAL command might look like:

   GLOBAL LOADLIB VSCSUSER VTAM VSCS NCP

8. Start VSCS with the command:

   VSCS START PARM=5

The VSCS START Command

To set up the environment in which VSCS operates, follow the instructions in "Defining Your Own Set of VSCS Start Options" on page 354 and "VTAM Considerations" on page 350.

After you have set up its operating environment, you can start VSCS with the VSCS START command.

Its format is:

VSCS START [PARM=x]

where *x* is a single digit from 0 to 9 that corresponds to a DTIUSERn load module in the VSCSUSER LOADLIB. If you leave out PARM=x, VSCS is started with the IBM-supplied set of default start options.

You can issue the VSCS START command:

- From the console of the VTAM and VSCS virtual machine
- From the PROFILE GCS for the VTAM and VSCS virtual machine.

The example in Figure 15 on page 48 shows VSCS being started by the PROFILE GCS.

If you use the VM Autolog facility in conjunction with the PROFILE GCS, you can start VSCS without operator intervention, when an IPL is done for the VM system. The examples in "Coding the PROFILEs for the Virtual Machines" on page 46 show the PROFILEs you can code to allow VSCS to be started automatically.

Normally the VSCS START command completes with a zero return code when VSCS is ready to accept additional commands. At this point you are free to issue VSCS and GCS commands from the console of the VTAM and VSCS virtual machine while initialization and execution are still going on. If VSCS initialization fails before reaching this point, you will get a non-zero return code.

You can start VSCS independently of VTAM. You can even start VSCS before VTAM. VSCS initialization will periodically retry start-up until VTAM is initialized and the VSCS application name has been activated. Retries occur every 30 seconds, and you can determine how long they will continue with the ACBLOOP operand on the DTIGEN macro.

Glossary

This glossary defines important NCP, NetView, SSP, and VTAM abbreviations and terms. It includes information from the *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699. Definitions from the *American National Dictionary for Information Processing* are identified by an asterisk (*). Definitions from draft proposals and working papers under development by the International Standards Organization, Technical Committee 97, Subcommittee 1 are identified by the symbol **(TC97)**. Definitions from the *CCIT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the Consultative Committee on International Telegraph and Telephone of the International Telecommunication Union, Geneva, 1980 are preceded by the symbol **(CCITT/ITU)**. Definitions from published sections of the *ISO Vocabulary of Data Processing*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1 and from published sections of the *ISO Vocabulary of Office Machines*, developed by subcommittees of ISO Technical Committee 95, are preceded by the symbol **(ISO)**.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

Reference Words Used in the Entries

The following reference words are used in this glossary:

Contrast with. Refers to a term that has an opposed or substantively different meaning.

Deprecated term for. Indicates that the term should not be used. It refers to a preferred term, which is defined.

See. Refers to multiple-word terms that have the same last word.

See also. Refers to related terms that have similar (but not synonymous) meanings.

Synonym for. Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

Synonymous with. Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

ABEND. Abnormal end of task.

abnormal end of task (ABEND). Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

ACB. (1) In VTAM, application control block. (2) In NCP, adapter control block.

ACB name. (1) The name of an ACB macro instruction. (2) A name specified in the ACBNAME parameter of a VTAM APPL statement. Contrast with *network name.*

accept. For a VTAM application program, to establish a session with a logical unit (LU) in response to a CINIT request from a system services control point (SSCP). The session-initiation request may begin when a terminal user logs on, a VTAM application program issues a macro instruction, or a VTAM operator issues a command. See also *acquire (1).*

access method. A technique for moving data between main storage and input/output devices.

accounting exit routine. In VTAM, an optional installation exit routine that collects statistics about session initiation and termination.

ACF. Advanced Communications Function.

ACF/NCP. Advanced Communications Function for the Network Control Program. Synonym for *NCP.*

ACF/TAP. Advanced Communications Function for the Trace Analysis Program. Synonym for *TAP.*

acquire. (1) For a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macro instruction. See also *accept.* (2) To take over resources that were formerly controlled by an access method in another domain, or to resume control of resources that were controlled by this domain but released. Contrast with *release.* See also *resource takeover.*

activate. To make a resource of a node ready to perform the functions for which it was designed. Contrast with *deactivate.*

active. (1) The state a resource is in when it has been activated and is operational. Contrast with *inactive, pending,* and *inoperative.* (2) Pertaining to a major or minor node that has been activated by VTAM. Most resources are activated as part of VTAM start processing or as the result of a VARY ACT command.

active application. The application subsystem currently in an extended recovery facility (XRF) session with a terminal user. See *alternate application.*

adapter control block (ACB). In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, start-stop lines, or SDLC links.

address translation. See *network address translation.*

adjacent SSCP table. A table containing lists of the system services control points (SSCPs) that VTAM can be in session with or can use to reach destination SSCPs in the same network or in other networks. The table is filed in the VTAM definition library.

Advanced Communications Function (ACF). A group of IBM program products (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

alias address. An address used by a gateway NCP and a gateway system services control point (SSCP) in one network to represent a logical unit (LU) or SSCP in another network.

alias name. A name defined in a host used to represent a logical unit name, logon mode table name, or class of service name in another network. This name is defined to a name translation program when the alias name does not match the real name. The alias name translation program is used to associate the real and alias names.

alias name translation facility. A function for converting logical unit names, logon mode table names, and class of service names used in one network into equivalent names to be used in another network. Available with NetView or NCCF program products.

alternate application. The subsystem that is prepared to take over a particular active application's extended recovery facility (XRF) sessions with terminal users in case the application fails. See *active application.*

API. Application program interface.

application control block (ACB). A control block that links an application program to VSAM or VTAM.

application program. (1) A program written for or by a user that applies to the user's work. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface (API). (1) The formally defined programming language interface between an IBM system control program or program product and its user. (2) The interface through which an application program interacts with an access method. In VTAM, it is the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

application program major node. A group of application program minor nodes. In the VTAM definition library, it is a member, book, or file that contains one or more APPL statements, which represent application programs. In MVS, it is a member of the library; in VSE, it is a book; and in VM, it is a CMS file of filetype VTAMLST.

ASCII. American National Standard Code for Information Interchange.

authorization exit routine. In VTAM, an optional installation exit routine that approves or disapproves requests for session initiation.

authorized path. In VTAM for MVS, a facility that enables an application program to specify that a data transfer or related operation be carried out in a privileged and more efficient manner.

automatic activation. In VTAM, the activation of links and link stations in adjacent subarea nodes as a result of channel device name or RNAME specifications related to an activation command that names a subarea node. See *direct activation.*

automatic logon. (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session will be between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. See also *controlling application program* and *controlling logical unit.* (2) In VM, a process by which a virtual machine is initiated by other than the user of that virtual machine. For example, the primary VM operator's virtual machine is activated automatically during VM initialization.

available. In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

back-level. Pertaining to an earlier release of an IBM product, which may not support a particular, current function.

BASE disk. The virtual disk that contains the text decks and macro instructions for VTAM and VM SNA console support (VSCS). It also contains control files and sample files used when running VTAM on the VM operating system. See *DELTA disk*, *MERGE disk*, *RUN disk*, and *ZAP disk.*

bidder. In SNA, the LU-LU half-session defined at session activation as having to request and receive permission from the other LU-LU half-session to begin a bracket. Contrast with *first speaker.* See also *bracket protocol.*

binary synchronous communication (BSC). (1) Communication using binary synchronous line discipline. (2) A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

bind. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request.* Contrast with *UNBIND.*

BIU segment. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either a request/response header (RH) followed by all or a portion of a request/response unit (RU), or only a portion of an RU.

blocking of PIUs. In SNA, an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

boundary function. In SNA: (1) A capability of a subarea node to provide protocol support for adjacent peripheral nodes, such as: (a) transforming network addresses to local addresses, and vice versa; (b) performing session sequence numbering for low-function peripheral nodes; and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *path control (PC) network* and *network addressable unit (NAU).*

boundary node. A subarea node that performs boundary functions. See also *boundary function.*

bracket protocol. In SNA, a data flow control protocol in which exchanges between the two LU-LU half-sessions are achieved through the use of brackets, with one LU designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder* and *first speaker.*

BSC. Binary synchronous communication.

buffer. A portion of storage for temporarily holding input or output data.

CCP. Configuration control program facility.

CCS. Console communication services.

CDRM. Cross-domain resource manager.

CDRSC. Cross-domain resource.

chain. See *RU chain.*

channel adapter. A communication controller hardware unit used to attach the controller to a System/360 or a System/370 channel.

channel-attached. Pertaining to the attachment of devices directly by data channels (I/O channels) to a host processor. Contrast with *link-attached.* Synonymous with *local-attached.*

channel-attachment major node. (1) A major node that includes an NCP that is channel-attached to a data host. (2) A major node that may include minor nodes that are the line groups and lines that represent a channel attachment to an adjacent (channel-attached) host. (3) In VM or VSE operating systems, a major node that may include minor nodes that are resources (host processors, NCPs, line groups, lines, SNA physical units and logical units, cluster controllers, and terminals) attached through a communication adapter.

character-coded. Synonym for *unformatted.*

CINIT. A network services request sent from a system services control point (SSCP) to a logical unit (LU) asking that LU to establish a session with another LU and to act as the primary end of the session.

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

cleanup. A network services request, sent by a system services control unit (SSCP) to a logical unit (LU), that causes a particular LU-LU session with that LU to be ended immediately and without the participation of either the other LU or its SSCP.

clear session. A session in which only clear data is transmitted or received. Contrast with *cryptographic session.*

cluster controller. A device that can control the input/output operations of more than one device connected to it. A cluster controller may be controlled by a program stored and executed in the unit; for example, the IBM 3601 Finance Communication Controller. Or it may be controlled entirely by hardware; for example, the IBM 3272 Control Unit.

CMC. Communication management configuration.

CMS. Conversational Monitor System.

CNM. Communication network management.

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in a FID4 transmission header. See also *VTAM operator command.*

communication adapter. An optional hardware feature, available on certain processors, that permits communication lines to be attached to the processors.

communication control unit. A communication device that controls the transmission of data over lines in a network. Communication control units include transmission control units (such as the 2702 Transmission Control Unit) and communication controllers (such as the 3705 or 3725).

communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit; for example, the IBM 3725 Communication Controller. It manages the details of line control and the routing of data through a network.

communication line. Deprecated term for *telecommunication line* and *transmission line.*

communication management configuration (CMC). (1) In VTAM, a technique for configuring a network that allows for the consolidation of many network management functions for the entire network in a single host processor. (2) A multiple-domain network configuration in which one of the hosts, called the communication management host, performs most of the controlling functions for the network, thus allowing the other hosts, called data hosts, to process applications. This is accomplished by configuring the network so that the communication management host owns most of the resources in the network that are not application programs. The resources that are not owned by the communication management host are the resources that are channel-attached stations of data hosts.

communication management host. The host processor in a communication management configuration that does all network-control functions in the network except for the control of devices channel-attached to data hosts. Contrast with *data host.*

communication network management (CNM). The process of designing, installing, operating, and managing the distribution of information and controls among end users of communication systems.

communication network management (CNM) application program. A VTAM application program that issues and receives formatted management services request units for physical units. For example, NetView.

communication network management (CNM) interface. The interface that the access method provides to an application program for handling data and commands associated with communication system management. CNM data and commands are handled across this interface.

communication network management (CNM) processor. A program that manages one of the functions of a communications system. A CNM processor is executed under control of NetView.

communication scanner processor (CSP). A processor in the 3725 Communication Controller that contains a microprocessor with control code. The code controls transmission of data over links attached to the CSP.

configuration. (1) (TC97) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

Configuration control program (CCP) facility. An SSP interactive application program facility by which configuration definitions for the IBM 3710 Network Controller can be created, modified, and maintained.

configuration restart. In VTAM, the recovery facility that can be used after a failure or deactivation of a major node, VTAM, or the host processor to restore the domain to its status at the time of the failure or deactivation.

configuration services. In SNA, one of the types of network services in the system services control point (SSCP) and in the physical unit (PU); configuration services activate, deactivate, and maintain the status of physical units, links, and link stations. Configuration services also shut down and restart network elements and modify path control routing tables and address-translation tables. See also *maintenance services*, *management services*, *network services*, *session services*, and *system services control point*.

connected. In VTAM, pertaining to a physical unit (PU) or logical unit (LU) that has an active physical path to the host processor containing the system services control point (SSCP) that controls the PU or LU.

connection. Synonym for *physical connection*.

console communications services (CCS). The SNA facility that acts as an interface between the control program and the VSCS component of VTAM for VM.

control block. (ISO) A storage area used by a computer program to hold control information.

control program (CP). The VM operating system that manages the real processor's resources and is responsible for simulating System/370s for individual users.

controlling application program. In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit*.

controlling logical unit. In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program*.

control statement. A statement in a command list that controls the processing sequence of the command list or allows the command list to send messages to the operator and receive input from the operator.

Conversational Monitor System (CMS). A VM application program for general interactive time sharing, problem solving, and program development.

converted command. An intermediate form of a character-coded command produced by VTAM through use of an unformatted system services definition table. The format of a converted command is fixed; the unformatted system services definition table must be constructed in such a manner that the character-coded command (as entered by a logical unit) is converted into the predefined, converted command format. See also *unformatted*.

COS. Class of service.

coupler. A hardware device that connects a modem to a public phone system in much the same way that a telephone does.

CP. Control program.

cross keys. Synonym for *cross-domain keys*.

cross-domain. In SNA, pertaining to control of resources involving more than one domain.

cross-domain keys. In SNA, a pair of cryptographic keys used by a system services control point (SSCP) to encipher the session cryptography key that is sent to another SSCP and to decipher the session cryptography key that is received from the other SSCP during initiation of cross-domain LU-LU sessions that use session-level cryptography. Synonymous with *cross keys.*

cross-domain link. (1) A subarea link connecting two subareas that are in different domains. (2) A link physically connecting two domains.

cross-domain resource (CDRSC). A resource owned by a cross-domain resource manager (CDRM) in another domain but known by the CDRM in this domain by network name and associated CDRM.

cross-domain resource manager (CDRM). In VTAM, the function in the system services control point (SSCP) that controls initiation and termination of cross-domain sessions.

cross-network. In SNA, pertaining to control or resources involving more than one SNA network.

cross-network LU-LU session. In SNA, a session between logical units (LUs) in different networks.

cross-network session. An LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

cryptographic. Pertaining to the transformation of data to conceal its meaning. See also *encipher* and *decipher.*

cryptographic key. In systems using the Data Encryption Standard (DES), a 64-bit value (containing 56 independent bits and 8 parity bits) provided as input to the algorithm in determining the output of the algorithm. See *cross-domain keys, session cryptography key, host master key,* and *secondary logical unit key.*

cryptographic session. In SNA products, an LU-LU session in which a function management data (FMD) request may be enciphered before it is transmitted and deciphered after it is received. Contrast with *clear session.* See *required cryptographic session* and *selective cryptographic session.*

CSP. Communication scanner processor.

data flow control (DFC) layer. In SNA, the layer within a half-session that (1) controls whether the half-session can send, receive, or concurrently send and receive request units (RUs); (2) groups related RUs into RU chains; (3) delimits transactions via the bracket protocol; (4) controls the interlocking of requests and responses in accordance with control modes specified at session activation; (5) generates sequence numbers; and (6) correlates requests and responses.

data host. In a communication management configuration, a host that is dedicated to processing applications and does not control network resources, except for its channel-attached or communication adapter-attached devices. Contrast with *communication management host.*

data link. In SNA, synonym for *link.*

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a link between two nodes and perform error control for the link. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

data link control protocol. In SNA, a set of rules used by two nodes on a data link to accomplish an orderly exchange of information. Synonymous with *line control.*

deactivate. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate.*

decipher. To convert enciphered data into clear data. Contrast with *encipher.* Synonymous with *decrypt.*

decrypt. To convert encrypted data into clear data. Contrast with *encrypt.* Synonym for *decipher.*

decryption. The unscrambling of data using an algorithm which works under the control of a key. The key allows data to be protected even when the algorithm is unknown. Data is unscrambled after transmission. Contrast with *encryption.*

default SSCP list. A list of system services control points (SSCPs), either in VTAM's network or another network, that can be used when no predefined cross-domain resource (CDRSC) or name translation function is provided specifying an LU's owning cross-domain resource manager (CDRM). This list is filed as a part of an adjacent SSCP table in the VTAM definition library.

default SSCP selection. A VTAM function that selects a set of one or more system services control points (SSCPs) to which a session request can be routed when there is no predefined cross-domain resource (CDRSC) or name translation function provided that specifies an LU's owning cross-domain resource manager (CDRM). See also *default SSCP list.*

definite response (DR). In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response* and *no response*.

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See also *macro instruction*.

DELTA disk. The virtual disk in a VM operating system that contains program temporary fixes (PTFs) that have been installed but not merged. See *BASE disk*, *MERGE disk*, *RUN disk*, and *ZAP disk*.

destination logical unit (DLU). The logical unit to which data is to be sent. Contrast with *origin logical unit (OLU)*.

dial-in. Refers to the direction in which a switched connection is requested by any node or terminal other than the receiving host or an NCP.

dial-out. Refers to the direction in which a switched connection is requested by a host or an NCP.

direct activation. In VTAM, the activation of a resource as a result of an activation command specifically naming the resource. See *automatic activation*. Contrast with *indirect activation*.

directory. In VM, a control program (CP) disk that defines each virtual machine's normal configuration.

disabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited*.

disconnection. The termination of a physical connection.

discontiguous shared segment. An area of virtual storage outside the address range of a virtual machine. It can contain read-only data or reentrant code. It connects discontiguous segments to a virtual machine's address space so programs can be fetched.

display. (1) To present information for viewing, usually on a terminal screen or a hard-copy device. (2) A device or medium on which information is presented, such as a terminal screen. (3) Deprecated term for *panel*.

DLU. Destination logical unit.

domain. (1) An access method, its application programs, communication controllers, connecting lines, modems, and attached terminals. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See also *single-domain network* and *multiple-domain network*.

domain operator. In a multiple-domain network, the person or program that controls the operation of the resources controlled by one system services control point. Contrast with *network operator* (2).

DR. (1) In NCP and CCP, dynamic reconfiguration. (2) In SNA, definite response.

DRDS. Dynamic reconfiguration data set.

dump. (1) Computer printout of storage. (2) To write the contents of all or part of storage to an external medium as a safeguard against errors or in connection with debugging. (3) (ISO) Data that have been dumped.

duplex. * In data communication, pertaining to a simultaneous two-way independent transmission in both directions. Synonymous with *full duplex*. Contrast with *half duplex*.

dynamic reconfiguration (DR). The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables.

dynamic reconfiguration data set (DRDS). In VTAM, a data set used for storing definition data that can be applied to a generated communication controller configuration at the operator's request. See also *dynamic reconfiguration*.

EBCDIC. * Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

element. (1) A field in the network address. (2) The particular resource within a subarea identified by the element address. See also *subarea*.

element address. In SNA, a value in the element address field of the network address identifying a specific resource within a subarea. See *subarea address*.

emulation mode. The function of a network control program that enables it to perform activities equivalent to those performed by a transmission control unit. Contrast with *network control mode.*

Emulation Program (EP). An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program.*

enabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is now ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *disabled* and *inhibited.*

encipher. (1) To scramble data or convert it, before transmission, to a secret code that masks the meaning of the data to any unauthorized recipient. (2) In VTAM, to convert clear data into enciphered data. Contrast with *decipher.* Synonymous with *encrypt.*

encrypt. Synonym for *encipher.*

encryption. The scrambling or encoding of data using an algorithm which works under the control of a key. The key allows data to be protected even when the algorithm is unknown. Data is scrambled prior to transmission. Contrast with *decryption.*

end user. In SNA, the ultimate source or destination of application data flowing through an SNA network. An end user may be an application program or a terminal operator.

EP. Emulation Program.

ER. (1) Explicit route. (2) Exception response.

ESTAE. Extended specify task abnormal exit.

event. (1) In NetView, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as an input/output operation.

exception response (ER). In SNA, a negative response shown as a value in the form-of-response-requested field of a request header (RH). An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response* and *no response.* See also *negative response.*

EXEC. In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

exit routine. Any of several types of special-purpose user-written routines. See *accounting exit routine, authorization exit routine, logon-interpret routine, virtual route selection exit routine, EXLST exit routine,* and *RPL exit routine.*

EXLST exit routine. In VTAM, a routine whose address has been placed in an exit list (EXLST) control block. The addresses are placed there with the EXLST macro instruction, and the routines are named according to their corresponding operand; hence DFASY exit routine, TPEND exit routine, RELREQ exit routine, and so forth. All exit list routines are coded by the VTAM application programmer. Contrast with *RPL exit routine.*

expedited flow. In SNA, a data flow designated in the transmission header (TH) that is used to carry network control, session control, and various data flow control request/response units (RUs); the expedited flow is separate from the normal flow (which carries primarily end-user data) and can be used for commands that affect the normal flow. Contrast with *normal flow.*

explicit route (ER). In SNA, the path control network elements, including a specific set of one or more transmission groups, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR).* See also *path* and *route extension.*

extended network addressing. The network addressing system that splits the address into an 8-bit subarea and a 15-bit element portion. The subarea portion of the address is used to address host processors or communication controllers. The element portion is used to permit processors or controllers to address resources.

extended recovery facility (XRF). Software designed to minimize the effect of failures in MVS, VTAM, the host processor, or IMS/VS on sessions between IMS/VS and designated terminals. It provides an alternate subsystem to take over failing sessions.

extended specify task abnormal exit (ESTAE). In VTAM, an MVS macro instruction that provides recovery capability and gives control to the user-specified exit routine for processing, diagnosing an ABEND, or specifying a retry address.

feature. A particular part of an IBM product that a customer can order separately.

field-formatted. Pertaining to a request or response that is encoded into fields, each having a specified format such as binary codes, bit-significant flags, and symbolic names. Contrast with *character-coded.*

first speaker. In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both half-sessions attempt to begin a bracket simultaneously. Contrast with *bidder.* See also *bracket protocol.*

flow control. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units, with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *pacing, session-level pacing,* and *virtual route pacing.*

formatted system services. A portion of VTAM that provides certain system services as a result of receiving a field-formatted command, such as an Initiate or Terminate command. Contrast with *unformatted system services (USS).* See also *field-formatted.*

frame. (1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

full duplex (FDX). Synonym for *duplex.*

full-screen mode. A form of panel presentation in NetView where the contents of an entire terminal screen can be displayed at once. Full-screen mode can be used for fill-in-the-blanks prompting. Contrast with *line mode.*

gateway. The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP.

gateway control functions. Functions performed by a gateway system services control point (SSCP) in conjunction with the gateway NCP to assign alias network address pairs for LU-LU sessions, assign virtual routes for the LU-LU sessions in adjacent networks, and translate network names within BIND RUs.

gateway host. A host node that contains a gateway system services control point (SSCP).

gateway NCP. An NCP that performs address translation to allow cross-network session traffic. The gateway NCP connects two or more independent SNA networks.

gateway SSCP. An SSCP that is capable of cross-network session initiation, termination, takedown, and session outage notification. A gateway SSCP is in session with the gateway NCP; it provides network name translation and assists the gateway NCP in setting up alias network addresses for cross-network sessions.

GCS. Group control system.

generation. The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

generation definition. The definition statement of a resource used in generating a program.

generic bind. Synonym for *session activation request.*

generic unbind. Synonym for *session deactivation request.*

group control system (GCS). A component of VM that provides multi-programming and shared memory support to virtual machines. It is a saved system intended for use with SNA products.

half-duplex. * In data communication, pertaining to an alternate, one way at a time, independent transmission. Contrast with *duplex.*

half-session. In SNA, a component that provides FMD services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also *primary half-session* and *secondary half-session.*

hard copy. A printed copy of machine output in a visually readable form; for example, printed reports, listings, documents, summaries, or network logs.

help panel. An online display that tells you how to use a command or another aspect of a product. See *task panel.*

High Performance Option (HPO). A program product that is an extension of VM/SP. It provides performance and operation enhancements for large system environments.

hierarchy. In NetView, the resource types, display types, and data types that make up the organization, or levels, in a network.

host master key. In SNA, deprecated term for *master cryptography key.*

host processor. (1) (TC97) A processor that controls all or part of a user application network. (2) In a network, the processing unit in which the data communication access method resides. (3) In an SNA network, the processing unit that contains a system services control point (SSCP).

HPO. High Performance Option.

IMR. Intensive mode recording.

inactive. In VTAM, describes the state of a resource that has not been activated or for which the VARY INACT command has been issued. Contrast with *active.* See also *inoperative.*

indirect activation. In VTAM, the activation of a lower-level resource of the resource hierarchy as a result of SCOPE or ISTATUS specifications related to an activation command naming a higher-level resource. Contrast with *direct activation.*

inhibited. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *enabled* and *disabled.*

initiate. A network services request sent from a logical unit (LU) to a system services control point (SSCP) requesting that an LU-LU session be established.

inoperative. The condition of a resource that has been active, but is not. The resource may have failed, received an INOP request, or is suspended while a reactivate command is being processed. See also *inactive.*

intensive mode recording (IMR). An NCP function that forces recording of temporary errors for a specified resource.

interactive problem control system (IPCS). A VM facility for diagnosing problems, and managing problem information and status. IPCS is the principal means for diagnosing virtual machine dumps.

interconnected networks. SNA networks connected by gateways.

interconnection. See *SNA network interconnection.*

interface. * A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

intermediate routing node (IRN). In SNA, a subarea node with intermediate routing function. A subarea node may be a boundary node, an intermediate routing node, both, or neither, depending on how it is used in the network.

intermediate SSCP. An SSCP along a session initiation path that owns neither of the LUs involved in a cross-network LU-LU session.

interpret table. In VTAM, an installation-defined correlation list that translates an argument into a string of eight characters. Interpret tables can be used to translate logon data into the name of an application program for which the logon is intended.

inter-user communication vehicle (IUCV). A VM facility for passing data between virtual machines and VM components.

IPCS. Interactive problem control system.

ISTATUS. In VTAM and NCP, a definition specification method for indicating the initial status of resources. See also *indirect activation.*

item. In CCP, any of the components, such as communication controllers, lines, cluster controllers, and terminals, that comprise an IBM 3710 Network Controller configuration.

IUCV. Inter-user communication vehicle.

JCL. Job control language.

job control language (JCL). * A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

keyword. (1) * One of the predefined words of an artificial language. (2) One of the significant and informative words in a title or document that describes the content of that document. (3) A symbol that identifies a parameter. (4) A part of a command operand that consists of a specific character string (such as DSNAME=).

line. See *communication line.*

line mode. A form of screen presentation in which the information is presented a line at a time in the message area of the terminal screen. Contrast with *full-screen mode.*

line control. Synonym for *data link control protocol.*

line group. One or more telecommunication lines of the same type that can be activated and deactivated as a unit.

link. In SNA, the combination of the link connection and the link stations joining network nodes; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with *data link.*

link level 2 test. See *link test.*

Link Problem Determination Aid (LPDA). A series of testing procedures initiated by NCP that provide modem status, attached device status, and the overall quality of a communications link.

link station. (1) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. (2) In VTAM, a named resource within a subarea node that represents another subarea node that is attached by a cross-subarea link. In the resource hierarchy, the link station is subordinate to the cross-subarea link.

link test. In SNA, a test in which one link station returns data received from another link station without changing the data in order to test the operation of the link. Three tests can be made; they differ in the resources that are dedicated during the test.

link-attached. In VTAM, pertaining to devices that are physically connected by a telecommunication line. Synonymous with *remote.* Contrast with *channel-attached.*

load module. (ISO) A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor.

local address. In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

local-attached. Deprecated term for *channel-attached.*

local non-SNA major node. In VTAM, a major node whose minor nodes are channel-attached non-SNA terminals.

local SNA major node. In VTAM, a major node whose minor nodes are channel-attached peripheral nodes.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other logical units. See also *network addressable unit (NAU), peripheral LU, physical unit (PU), system services control point (SSCP), primary logical unit (PLU),* and *secondary logical unit (SLU).* Contrast with *physical unit (PU).*

logical unit (LU) services. In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also *physical unit (PU) services.*

logoff. In VTAM, an unformatted session termination request.

log on. (1) To initiate a session. (2) In SNA, to initiate a session between an application program and a logical unit (LU).

logon. In VTAM, an unformatted session initiation request for a session between two logical units. See *automatic logon* and *simulated logon.* See also *session-initiation request.*

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters.*

logon mode table. In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name.

logon-interpret routine. In VTAM, an installation exit routine, associated with an interpret table entry, that translates logon information. It may also verify the logon.

loop adapter. A feature of the IBM 4300 Processor family that allows the attachment of a variety of SNA and non-SNA devices. To VTAM, these devices appear as channel-attached type 2 physical units (PUs).

LPDA. Link Problem Determination Aid.

LU. Logical unit.

LU connection test. In SNA products, a diagnostic aid that permits a terminal operator to check whether the path between a system services control point (SSCP) and a logical unit (LU) is operational.

LU type. A deprecated term for *LU-LU session type.*

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

LU-LU session type. In SNA, the classification of an LU-LU session in terms of the specific subset of SNA protocols and options supported by the logical units (LUs) for that session, namely:

> The mandatory and optional values allowed in the session activation request.
>
> The usage of data stream controls, FM headers, request unit (RU) parameters, and sense codes.
>
> Presentation services protocols such as those associated with FM header usage.

LU-LU session types 0, 1, 2, 3, 4, 6, and 7 are defined.

macro instruction. (1) * (ISO) An instruction in a source language that is to be replaced by a defined sequence of instructions in the same source language. The macro instruction may also specify values for parameters in the instructions that are to replace it. (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macro instruction in the program. See also *definition statement.*

maintain system history program (MSHP). A program that facilitates the process of installing and servicing a VSE system.

maintenance and operator subsystem (MOSS). A subsystem of the 3725 Communication Controller that contains a processor and operates independently of the rest of the controller. It loads and supervises the 3725, runs problem determination procedures, and assists in maintaining both hardware and software.

maintenance services. In SNA, one of the types of network services in system services control points (SSCPs) and physical units (PUs). Maintenance services provide facilities for testing links and nodes and for collecting and recording error information. See also *configuration services, management services, network services,* and *session services.*

major node. In VTAM, a set of resources that can be activated and deactivated as a group. See *node* and *minor node.*

management services. In SNA, one of the types of network services in system services control points (SSCPs) and logical units (LUs). Management services forward requests for network data, such as error statistics, and deliver the data in reply. See also *configuration services, maintenance services, network services,* and *session services.*

mandatory cryptographic session. Synonym for *required cryptographic session.*

master cryptography key. In SNA, a cryptographic key used to encipher operational keys that will be used at a node.

maximum SSCP rerouting count. The maximum number of times a session initiation request will be rerouted to intermediate system services control points (SSCPs) before the request reaches the destination SSCP. This count is used to prevent endless rerouting of session initiation requests.

MERGE disk. The virtual disk in the VM operating system that contains program temporary fixes (PTFs) after the VMFMERGE EXEC is invoked. See *BASE disk, DELTA disk, RUN disk,* and *ZAP disk.*

message. In VTAM, the amount of FM data transferred to VTAM by the application program with one SEND request.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

minidisk. Synonym for *virtual disk.*

minor node. In VTAM, a uniquely-defined resource within a major node. See *node* and *major node.*

MOSS. Maintenance and operator subsystem.

MSHP. Maintain system history program.

multiple-domain network. In SNA, a network with more than one system services control point (SSCP). Contrast with *single-domain network.*

Multiple Virtual Storage (MVS). An IBM program product whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

Multiple Virtual Storage for Extended Architecture (MVS/XA). An IBM program product whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for Extended Architecture. Extended architecture allows 31-bit storage addressing. MVS/XA is a software operating system controlling the execution of programs.

MVS. Multiple Virtual Storage operating system.

MVS/XA. Multiple Virtual Storage for Extended Architecture operating system.

name translation. In SNA network interconnection, converting logical unit names, logon mode table names, and class of service names used in one network into equivalent names to be used in another network. This function can be provided through NetView and invoked by a gateway system services control point (SSCP) when necessary. See also *alias name.*

native network. The network in which a gateway NCP's resources reside.

NAU. Network addressable unit.

NCCF. Network Communications Control Facility.

NCP. (1) Network Control Program (IBM program product). Its full name is Advanced Communications Function for the Network Control Program. (2) Network control program (general term).

NCP/EP definition facility (NDF). A program that is part of System Support Programs (SSP) and is used to generate a partitioned emulation programming (PEP) load module or a load module for a Network Control Program (NCP) or for an Emulation Program (EP).

NCP major node. In VTAM, a set of minor nodes representing resources, such as lines and peripheral nodes, controlled by a network control program. See *major node.*

NCP Subset. Advanced Communications Function for Network Control Program (NCP) V4 Subset. An IBM licensed program that is a subset of NCP. It operates only on IBM 3720 Communication Controllers with certain capacity limitations such as number of scanners, lines, and channel adapters supported.

NCP/Token-Ring interconnection (NTRI). An NCP function that allows a communication controller to attach to the IBM Token-Ring Network by providing a basic boundary network node interface.

negative response. In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response.* See *exception response.*

NetView. An IBM program product used to monitor a network, manage it, and diagnose its problems.

network. (1) (TC97) An interconnected group of nodes. (2) In data processing, a user application network. See *path control network, public network, SNA network*, and *user application network.*

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address.* See also *network name.*

network address translation. In SNA network interconnection, conversion of the network address assigned to a logical unit in one network into an address in an adjacent network. This function is provided by the gateway NCP that joins the two networks. See also *alias network address* and *real network address.*

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*, and *path control network.*

Network Communications Control Facility (NCCF). (1) An IBM program product that is a base for command processors that can monitor, control, and improve the operations of a network. Its function is included and enhanced in NetView's command facility. (2) A traditional, alternative name for the command facility of NetView.

network control (NC). In SNA, an RU category used for requests and responses exchanged between physical units (PUs) for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjacent peripheral nodes. See also *data flow control layer* and *session control.*

network control mode. The functions of a network control program that enable it to direct a communication controller to perform activities such as polling, device addressing, dialing, and answering. Contrast with *emulation mode.*

Network Control Program (NCP). An IBM program product that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network control program. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

network controller. A concentrator and protocol converter used with SDLC links. By converting protocols, which manage the way data is sent and received, the IBM 3710 Network Controller allows the use of non-SNA devices with an SNA host processor.

network identifier (network ID). The network name defined to NCPs and hosts to indicate the name of the network in which they reside. It is unique across all communicating SNA networks.

networking. In a multiple-domain network, communication among domains.

Network Logical Data Manager (NLDM). (1) An IBM program product that collects and correlates session-related data and provides online access to this information. It runs as an NCCF communication network management (CNM) application program. Its function is included and enhanced in NetView's session monitor. (2) A traditional, alternative name for the session monitor of NetView.

network name. (1) In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link, or a link station. See also *network address.* (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program is its network name and it must be unique across domains. Contrast with *ACB name.* See *uninterpreted name.*

network node. Synonym for *node.*

network operator. (1) A person or program responsible for controlling the operation of all or part of a network. (2) The person or program that controls all the domains in a multiple-domain network. Contrast with *domain operator.*

Network Routing Facility (NRF). An IBM program product that resides in the NCP, which provides a path for messages between terminals, and routes messages over this path without going through the host processor.

network services (NS). In SNA, the services within network addressable units (NAUs) that control network operation through SSCP-SSCP, SSCP-PU, and SSCP-LU sessions. See *configuration services, maintenance services, management services,* and *session services.*

Network Terminal Option (NTO). An IBM program product that allows certain non-SNA devices to participate in sessions with SNA application programs in the host processor. NTO converts non-SNA protocol to SNA protocol when data is sent to the host from a non-SNA device and reconverts SNA protocol to non-SNA protocol when data is sent back to the device.

NIB. Node initialization block.

NLDM. Network Logical Data Manager.

node. (1) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. (2) In VTAM, a point in a network defined by a symbolic name. Synonymous with *network node.* See *major node* and *minor node.*

node initialization block (NIB). In VTAM, a control block associated with a particular node or session that contains information used by the application program to identify the node or session and to indicate how communication requests on a session are to be handled by VTAM.

node name. In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Four types are defined: 1, 2, 4, and 5. Type 1 and type 2 nodes are also referred to as peripheral nodes and type 4 and type 5 nodes are also referred to as subarea nodes. See also *physical unit type*.

non-native network. Any network attached to a gateway NCP that does not contain that NCP's resources.

nonswitched line. A telecommunication line on which connections do not have to be established by dialing. Contrast with *switched line*.

no response (NR). In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

normal flow. In SNA, a data flow designated in the transmission header (TH) that is used primarily to carry end-user data. The rate at which requests flow on the normal flow can be regulated by session-level pacing. Normal and expedited flows move in both the primary-to-secondary and secondary-to-primary directions. Contrast with *expedited flow*.

NPSI. X.25 NCP Packet Switching Interface.

NTO. Network Terminal Option.

OLU. Origin logical unit.

operator. A person who operates a machine. See *network operator*.

operator profile. In NetView, the resources and activities a network operator has control over. The statements defining these resources and activities are stored in a file that is activated when the operator logs on.

origin logical unit (OLU). The logical unit from which data is sent. Contrast with *destination logical unit (DLU)*.

pacing. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See *session-level pacing, send pacing*, and *virtual route (VR) pacing*. See also *flow control*.

pacing group. In SNA, (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. Synonymous with *window*. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests.

pacing group size. In SNA, (1) The number of path information units (PIUs) in a virtual route pacing group. The pacing group size varies according to traffic congestion along the virtual route. Synonymous with *window size*. (2) The number of requests in a session-level pacing group.

pacing response. In SNA, an indicator that signifies a receiving component's readiness to accept another pacing group; the indicator is carried in a response header (RH) for session-level pacing, and in a transmission header (TH) for virtual route pacing.

packet switching. (TC97) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet; upon completion of the transmission, the channel is made available for the transfer of other packets.

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also *scroll*. (3) (ISO) In a virtual storage system, a fixed-length block that has a virtual address and that can be transferred between real storage and auxiliary storage. (4) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. See also *help panel* and *task panel*. Contrast with *screen*. (2) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

parallel sessions. In SNA, two or more concurrently active sessions between the same two logical units (LUs) using different pairs of network addresses. Each session can have independent session parameters.

partitioned emulation programming (PEP) extension. A function of a network control program that enables a communication controller to operate some telecommunication lines in network control mode while simultaneously operating others in emulation mode.

path. (1) In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). A path consists of a virtual route and its route extension, if any. See also *explicit route.* (2) In defining a switched major node, a potential dial-out port that can be used to reach a physical unit.

path control (PC) layer. In SNA, the layer that manages the sharing of link resources of the SNA network and routes basic information units (BIUs) through it. Path control routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units (BTUs) and one or more PIUs with data link control. See also *BIU segment, blocking of PIUs, data link control layer,* and *transmission control layer.*

path control (PC) network. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network* and *user application network.* See also *boundary function.*

path information unit (PIU). In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header.*

PC. Path control.

PEP. Partitioned emulation programming.

peripheral LU. In SNA, a logical unit representing a peripheral node.

peripheral node. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary function assistance from an adjacent subarea node. A peripheral node is a type 1 or type 2 node connected to a subarea node.

peripheral PU. In SNA, a physical unit representing a peripheral node.

physical connection. In VTAM, a point-to-point connection or multipoint connection.

physical unit (PU). In SNA, one of three types of network addressable units (NAUs). Each node of an SNA network contains a physical unit (PU) that manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) via an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. See also *peripheral PU, physical unit (PU) type,* and *subarea PU.*

physical unit (PU) services. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also *logical unit (LU) services.*

physical unit (PU) type. In SNA, the classification of a physical unit (PU) according to the type of node in which it resides. The PU type is the same as its node type; that is, a type 1 PU resides in a type 1 node, and so forth.

PIU. Path information unit.

PLU. Primary logical unit.

polling. (1) * Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (2) (TC97) The process whereby stations are invited, one at a time, to transmit.

positive response. A response indicating that a request was received and processed. Contrast with *negative response.*

PPT. Primary POI task.

primary application program. In VTAM, an application program acting as the primary end of an LU-LU session.

primary end of a session. The end of a session that uses primary protocols. The primary end establishes the session. For an LU-LU session, the primary end of the session is the primary logical unit. Contrast with *secondary end of a session.* See *half-session.*

primary half-session. In SNA, the half-session that sends the session activation request. See also *primary logical unit.* Contrast with *secondary half-session.*

primary logical unit (PLU). In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU).*

primary POI task (PPT). The NetView subtask that processes all unsolicited messages received from the VTAM program operator interface (POI) and delivers them to the controlling operator or to the command processor. The PPT also processes the initial command specified to execute when NetView is initialized and timer request commands scheduled to execute under the PPT.

problem determination. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environment failure such as a power loss, or a user error.

profile. In the Conversational Monitor System (CMS) or the group control system (GCS), the characteristics defined by a PROFILE EXEC file that executes automatically after the system is loaded into a virtual machine. See also *operator profile.*

programmed operator. A VTAM application program that is authorized to issue VTAM operator commands and receive VTAM operator awareness messages. See also *solicited messages* and *unsolicited messages.*

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM in a current unaltered release of the program.

protection key. An indicator that appears in the current program status word whenever an associated task has control of the system. This indicator must match the storage keys of all main storage locks that the task is to use.

protocol. (1) (CCITT/ITU) A specification for the format and relative timing of information exchanged between communicating parties. (2) (TC97) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. See also *bracket protocol.* Synonymous with *line control discipline* and *line discipline.* See also *link protocol.*

PTF. Program temporary fix.

PU. Physical unit.

PU type. Physical unit type.

public network. A network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. Contrast with *user-application network.*

PU-PU flow. In SNA, the exchange between physical units (PUs) of network control requests and responses.

real name. The name by which a logical unit (LU), logon mode table, or class of service (COS) table is known within the SNA network in which it resides.

real network address. The address by which a logical unit (LU) is known within the SNA network in which it resides.

receive pacing. In SNA, the pacing of message units that the component is receiving. See also *send pacing.*

Recommendation X.21 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for a general purpose interface between data terminal equipment and data circuit equipment for synchronous operations on a public data network.

Recommendation X.25 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching.*

release. For VTAM to relinquish control of resources (communication controllers or physical units). See also *resource takeover.* Contrast with *acquire (2).*

remote. Synonym for *link-attached.*

remote spooling communications subsystem (RSCS). A VM networking component that provides telecommunication facilities for the transmission of bulk files between VM users and remote stations.

request header (RH). In SNA, control information preceding a request unit (RU). See also *request/response header (RH).*

request parameter list (RPL). In VTAM, a control block that contains the parameters necessary for processing a request for data transfer, for establishing or terminating a session, or for some other operation.

request unit (RU). In SNA, a message unit that contains control information such as a request code or FM headers, end-user data, or both.

request/response header (RH). In SNA, control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

request/response unit (RU). In SNA, a generic term for a request unit or a response unit. See also *request unit (RU)* and *response unit.*

required cryptographic session. A cryptographic session in which all outbound data is enciphered and all inbound data is deciphered. Synonymous with *mandatory cryptographic session.* Contrast with *selective cryptographic session* and *clear session.*

resource. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In NetView, any hardware or software that provides function to the network.

resource takeover. In VTAM, action initiated by a network operator to transfer control of resources from one domain to another. See also *acquire (2)* and *release.* See *takeover.*

response header (RH). In SNA, a header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response, pacing response*, and *positive response.*

response time. (1) The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal. (2) For response time monitoring, the time from the activation of a transaction until a response is received, according to the response time definition coded in the performance class.

response unit (RU). In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

return code. * A code [returned from a program] used to influence the execution of succeeding instructions.

RH. Request/response header.

ring. A network configuration where a series of attaching devices are connected by unidirectional transmission links to form a closed path.

route. See *explicit route* and *virtual route.*

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path, explicit route (ER), virtual route (VR).*

RPL. Request parameter list.

RPL exit routine. In VTAM, an application program exit routine whose address has been placed in the EXIT field of a request parameter list (RPL). VTAM invokes the routine to indicate that an asynchronous request has been completed. See *EXLST exit routine.*

RSCS. Remote spooling communications subsystem.

RU. Request/response unit.

RU chain. In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated via control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

RUN disk. The virtual disk that contains the VTAM and VM SNA console support (VSCS) load libraries, program temporary fixes (PTFs) and user-written modifications from the ZAP disk. See *BASE disk, DELTA disk, MERGE disk*, and *ZAP disk.*

scanner interface trace (SIT). A record of the activity within the communication scanner processor (CSP) for a specified data link between a 3725 Communication Controller and a resource.

SCIP exit. Session control in-bound processing exit.

screen. An illuminated display surface; for example, the display surface of a CRT or plasma panel. Contrast with *panel.*

scroll. To move all or part of the display image vertically to display data that cannot be observed within a single display image. See also *page (2).*

SDLC. Synchronous Data Link Control.

secondary application program. An application program acting as the secondary end of an LU-LU session.

secondary end of a session. That end of a session that uses secondary protocols. For an LU-LU session, the secondary end of the session is the secondary logical unit (SLU). Contrast with *primary end of a session.* See also *secondary logical unit (SLU)* and *half-session.*

secondary half-session. In SNA, the half-session that receives the session-activation request. See also *secondary logical unit (SLU).* Contrast with *primary half-session.*

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU).*

secondary logical unit (SLU) key. A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

selective cryptographic session. A cryptographic session in which an application program is allowed to specify the request units to be enciphered. Contrast with *required cryptographic session* and *clear session.*

send pacing. In SNA, pacing of message units that a component is sending. See also *receive pacing.*

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session. See *half-session, LU-LU session, SSCP-LU session, SSCP-PU session,* and *SSCP-SSCP session.* See also *LU-LU session type* and *PU-PU flow.*

session activation request. In SNA, a request that activates a session between two network addressable units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Synonymous with *generic BIND.* Contrast with *session deactivation request.*

session control (SC). In SNA, (1) One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification. (2) A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

session control in-bound processing exit (SCIP). A user exit that receives control when certain request units (RUs) are received by VTAM.

session cryptography key. In SNA, a data encrypting key used to encipher and decipher function management data (FMD) requests transmitted in an LU-LU session that uses cryptography.

session deactivation request. In SNA, a request that deactivates a session between two network addressable units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind.* Contrast with *session activation request.*

session limit. (1) In SNA, the maximum number of concurrently active LU-LU sessions a particular logical unit can support. (2) In the network control program, the maximum number of concurrent line-scheduling sessions on a non-SDLC, multipoint line.

session management exit routine. An installation-supplied VTAM exit routine that performs authorization, accounting, and gateway path selection functions.

session parameters. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units. See also *logon mode.*

session partner. In SNA, one of the two network addressable units (NAUs) having an active session.

session services. In SNA, one of the types of network services in the system services control point (SSCP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that the SSCP initiate or terminate sessions between logical units. See *configuration services* and *maintenance services.*

session-initiation request. In SNA, an Initiate or logon request from a logical unit (LU) to a system services control point (SSCP) that an LU-LU session be activated.

session-level pacing. In SNA, a flow control technique that permits a receiving connection point manager to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing* and *virtual route pacing.*

shadow resource. In VTAM, an alternate representation of a network resource that is retained as a definition for possible future use.

shared. Pertaining to the availability of a resource to more than one use at the same time.

simulated logon. A session-initiation request generated when a VTAM application program issues a SIMLOGON macro instruction. The request specifies a logical unit (LU) with which the application program wants a session in which the requesting application program will act as the primary logical unit (PLU).

single-domain network. In SNA, a network with one system services control point (SSCP). Contrast with *multiple-domain network.*

SIT. Scanner interface trace.

SLU. Secondary logical unit.

SMF. System management facility.

SMP. System Modification Program.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNA network interconnection. The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

SNA terminal. A terminal that supports Systems Network Architecture protocols.

span. In NetView, a user-defined group of network resources within a single domain. Each major or minor node is defined as belonging to one or more spans. See also *span of control.*

span of control. The total network resources over which a particular network operator has control. All the network resources listed in spans associated through profile definition with a particular network operator are within that operator's span of control.

SSCP. System services control point.

SSCP rerouting. In SNA network interconnection, the technique used by the gateway system services control point (SSCP) to send session-initiation request units (RUs), by way of a series of SSCP-SSCP sessions, from one SSCP to another, until the owning SSCP is reached.

SSCP-LU session. In SNA, a session between a system services control point (SSCP) and a logical unit (LU); the session enables the LU to request the SSCP to help initiate LU-LU sessions.

SSCP-PU session. In SNA, a session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

SSCP-SSCP session. In SNA, a session between the system services control point (SSCP) in one domain and the SSCP in another domain. An SSCP-SSCP session is used to initiate and terminate cross-domain LU-LU sessions.

SSP. System Support Programs (IBM program product). Its full name is Advanced Communications Function for System Support Programs.

ST. Session configuration screen abbreviation.

start option. In VTAM, a user-specified or IBM-supplied option that determines certain conditions that are to exist during the time a VTAM system is operating. Start options can be predefined or specified when VTAM is started.

station. (1) One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. (2) One or more computers, terminals, or devices at a particular location.

subarea (SA). A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea address. In SNA, a value in the subarea field of the network address that identifies a particular subarea. See also *element address*.

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See also *intermediate routing node, peripheral node*, and *node type*.

subarea PU. In SNA, a physical unit (PU) in a subarea node.

subsystem. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

supervisor. The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations.

supervisor call (SVC). A request that serves as the interface into operating system functions, such as allocating storage. The SVC protects the operating system from inappropriate user entry. All operating system requests must be handled by SVCs.

supervisor call (SVC) instruction. An instruction that interrupts the program being executed and passes control to the supervisor so that it can perform a specific service indicated by the instruction.

SVC. (1) Supervisor call. (2) With X.25 NPSI, switched virtual circuit.

switched line. A communication line in which the connection between the communication controller and a remote link station is established by dialing.

switched major node. In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

switched network backup (SNBU). In VTAM, an optional facility that allows a user to specify, for certain types of PUs, a switched line to be used as an alternate path if the primary line becomes unavailable or unusable.

switched virtual circuit (SVC). An X.25 NPSI circuit that is dynamically established when needed. The X.25 equivalent of a switched line.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute and High-Level Data Link Control (HDLC) of the International Standards Organization.

system management facility (SMF). A standard feature of MVS that collects and records a variety of system and job-related information.

System Modification Program (SMP). An operating system component that facilitates the process of installing and servicing an MVS system. See also *System Modification Program Extended.*

System Modification Program Extended (SMP/E). An IBM program product that facilitates the process of installing and servicing an MVS system. See also *System Modification Program.*

system services control point (SSCP). In SNA, a focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

System Support Programs (SSP). An IBM program product, made up of a collection of utilities and small programs, that supports the operation of the NCP.

takeover. The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem. See *resource takeover*.

TAP. Trace analysis program.

task. A basic unit of work to be accomplished by a computer. The task is usually specified to a control program in a multiprogramming or multiprocessing environment.

task panel. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

TCAM. (1) Telecommunications Access Method. (2) The IBM program product whose full name is Advanced Communications Function for TCAM and that provides queued message handling. TCAM Versions 1 and 2 are access methods, but TCAM Version 3 is a message handling subsystem.

TCAS. Terminal control address space.

telecommunication line. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

teletypewriter exchange service (TWX). Teletypewriter service in which suitably arranged teletypewriter stations are provided with lines to a central office for access to other such stations throughout the U.S. and Canada. Both baudot and ASCII-coded machines are used. Business machines may also be used with certain restrictions.

terminal. A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

terminal control address space (TCAS). The part of TSO/VTAM that provides logon services for TSO/VTAM users.

terminal component. An addressable part of a terminal that performs an input or output function, such as the display component of a keyboard-display device or a printer component of a keyboard-printer device.

terminate. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

TG. Transmission group.

TH. Transmission header.

threshold. In NetView, refers to a percentage value set for a resource and compared to a calculated error-to-traffic ratio.

TIC. Token-ring interface coupler.

time sharing option (TSO). An optional configuration of the operating system that provides conversational time sharing from remote stations.

time sharing option for VTAM (TSO/VTAM). An optional configuration of the operating system that provides conversational time sharing from remote stations in a network using VTAM.

token. A sequence of bits passed from one device to another along the network. When the token has data appended to it, it becomes a frame.

token ring. A network, having a ring topology, that passes tokens from one attaching device to another. For example, the IBM Token-Ring Network.

token-ring interface coupler (TIC). An adapter that can connect a 3725 Communication Controller to an IBM Token-Ring Network.

trace analysis program (TAP). An SSP program service aid that assists in analyzing trace data produced by VTAM, TCAM, and NCP and provides network data traffic and network error reports.

transmission control (TC) layer. In SNA, the layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and deciphers end-user data. Transmission control has two components: the connection point manager and session control. See also *half-session*.

transmission group (TG). In SNA, a group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages. A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

transmission header (TH). In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit.*

transmission line. Synonym for *telecommunication line.*

transmission priority. In SNA, a rank assigned to a path information unit (PIU) that determines its precedence for being selected by the transmission group control component of path control for forwarding to the next subarea node of the route used by the PIU.

TSO. Time sharing option.

TSO/VTAM. Time sharing option for VTAM.

TWX. Teletypewriter exchange service.

unbind. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request.* Contrast with *BIND.*

unformatted. In VTAM, pertaining to commands (such as LOGON or LOGOFF) entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user's unformatted system services definition table. Synonymous with *character-coded.* Contrast with *field-formatted.*

unformatted system services (USS). In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as a logon or logoff request into a field-formatted request for processing by formatted system services and translates field-formatted replies and responses into character-coded requests for processing by a logical unit. Contrast with *formatted system services.* See also *converted command.*

uninterpreted name. In SNA, a character string that a system services control point (SSCP) is able to convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

user. Anyone who requires the services of a computing system.

user exit. A point in an IBM-supplied program at which a user exit routine may be given control.

USERVAR. An application name used to route a session-establishment request to the currently active application subsystem.

user-application network. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. Contrast with *public network.*

USS. Unformatted system services.

variable. In NetView, a character string beginning with & that is coded in a command list and is assigned a value during execution of the command list.

virtual disk. (1) A logical subdivision (or all) of a physical disk pack in the VM operating system that has its own virtual device address, consecutive virtual cylinders, and a volume table of contents (VTOC) or disk label identifier. (2) Synonymous with *minidisk.*

virtual machine. A functional simulation of a computer and its associated devices.

Virtual Machine (VM). A program product whose full name is the Virtual Machine/System Product (VM/SP). It is a software operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

virtual route (VR). In SNA, a logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intra-node sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER), path,* and *route extension.*

virtual route (VR) pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing* and *session-level pacing.*

virtual route selection exit routine. In VTAM, an optional installation exit routine that modifies the list of virtual routes associated with a particular class of service before a route is selected for a requested LU-LU session.

virtual storage. (ISO) The notion of storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations.

Virtual Storage Access Method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

Virtual Storage Extended (VSE). An IBM program product whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs.

Virtual Telecommunications Access Method (VTAM). An IBM program product that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VM. Virtual Machine operating system. Its full name is Virtual Machine/System Product.

VM SNA console support (VSCS). A VTAM component for the VM environment that provides System Network Architecture (SNA) support. It allows SNA terminals to be virtual machine consoles. See also *VM/VTAM Communication Network Application.*

VM/SP. Virtual Machine/System Product operating system. Synonym for *VM.*

VR. Virtual route.

VSAM. Virtual Storage Access Method.

VSCS. VM SNA console support.

VSE. Virtual Storage Extended operating system.

VSE/AF. Virtual Storage Extended/Advanced Function operating system. Synonym for *VSE.*

VTAM. Virtual Telecommunications Access Method (IBM program product). Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method.

VTAM application program. A program that has opened an ACB to identify itself to VTAM and can now issue VTAM macro instructions.

VTAM definition. The process of defining the user application network to VTAM and modifying IBM-defined characteristics to suit the needs of the user.

VTAM definition library. The operating system files or data sets that contain the definition statements and start options filed during VTAM definition.

VTAM internal trace (VIT). A trace used in VTAM to collect data on channel I/O, use of locks, and storage management services.

VTAM operator. A person or program authorized to issue VTAM operator commands. See *domain operator, program operator,* and *network operator (2).*

VTAM operator command. A command used to monitor or control a VTAM domain.

VTAM Terminal I/O Coordinator (VTIOC). The part of TSO/VTAM that converts TSO TGET, TPUT, TPG, and terminal control macro instructions into SNA request units.

VTIOC. VTAM Terminal I/O Coordinator.

window. (1) In SNA, synonym for *pacing group.* (2) A small amount of information in a framed-in area on a panel that overlays part of the panel.

window size. In SNA, synonym for *pacing group size.*

XID. A data link control command and response passed between adjacent nodes that allows the two nodes to exchange identification and other information necessary for operation over the data link.

XRF. Extended recovery facility.

X.21. See *Recommendation X.21 (Geneva 1980).*

X.25. See *Recommendation X.25 (Geneva 1980).*

X.25 NCP Packet Switching Interface (NPSI). The X.25 Network Control Program Packet Switching Interface, which is an IBM program product that allows SNA users to communicate over packet-switched data networks that have interfaces complying with Recommendation X.25 (Geneva 1980) of the International Telegraph and Telephone Consultative Committee (CCITT). It allows SNA programs to communicate with SNA equipment or with non-SNA equipment over such networks. In addition, this product may be used to attach native X.25 equipment to SNA host systems without a packet network. See also *Recommendation X.25 (Geneva 1980).*

ZAP disk. The virtual disk in the VM operating system that contains the user-written modifications to VTAM code. See *BASE disk, DELTA disk, MERGE disk,* and *RUN disk.*

31-bit storage addressing. The storage address structure available in an MVS/XA operating system.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

U

V

W

X

Z

Numerics

VTAM
Version 3 Releases 1 and 1.1
Installation and Resource Definition

Publication No. SC23-0111-3

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are:

Clarity   Accuracy   Completeness   Organization   Coding   Retrieval   Legibility

If you wish a reply, give your name, company, mailing address, and date:

____
____
____
____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

SC23-0111-3

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E12
P.O. Box 12195
Research Triangle Park, N.C. 27709-2195

Fold and tape | Please Do Not Staple | Fold and tape

IBM ®

SC23-0111-3

Printed in U.S.A.

SC23-0111-03