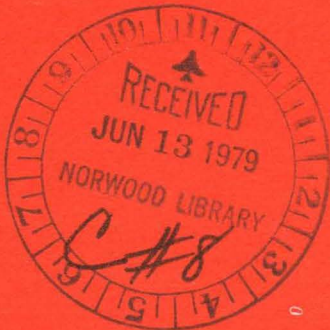


## Systems

## VTAM Concepts and Planning

Virtual Telecommunications  
Access Method (VTAM)  
VTAM Level 2

DOS/VS  
OS/VS1  
OS/VS2 SVS  
OS/VS2 MVS



*Includes: GX 27-1545  
GX31-0606  
GX31-0890*

# IBM

## **Systems**

## **VTAM Concepts and Planning**

**Virtual Telecommunications  
Access Method (VTAM)**

**VTAM Level 2**

**DOS/VS  
OS/VS1  
OS/VS2 SVS  
OS/VS2 MVS**



#### **Fourth Edition (June 1976)**

This edition, as amended by Technical Newsletters GN27-1545, GN31-0606 and GN31-0890, corresponds to VTAM Level 2 in

DOS/VS Releases 32, 33 and 34

OS/VS1 Releases 5, 6.0 and subsequent releases, if any

Component release to OS/VS2 SVS Release 1.7

OS/VS2 MVS Release 3.7 and subsequent releases, if any

The changes made for this edition and its subsequent newsletters are described in the Summary of Amendments.

Changes are continually made to the information in IBM system publications. Before using this publication in connection with the operation of IBM systems, consult your IBM representative or the *IBM System/370 Bibliography*, GC20-0001, to find out which editions are applicable and current.

Copies of this and other IBM publications can be obtained through IBM branch offices.

A form has been provided at the back of this publication for readers' comments. Address additional comments to IBM Corporation, Department 63T, Neighborhood Road, Kingston, New York 12401. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligations whatever. You may, of course, continue to use the information you supply.

**Summary of Amendments (December 15, 1978)  
to GC27-6998-3 by TNL GN31-0890**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS and OS/VS2 MVS**

**Changed Documentation** The edition notice has been updated. Other changes update publication references and clarify technical descriptions.

**Summary of Amendments (May 31, 1977)  
to GC27-6998-3 by TNL GN31-0606**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS  
Announcement of Additional 3270 Devices  
DOS/VS Release 34**

**New Program Features**

**Support of New 3270 Devices:** The new 3270 devices that are supported by VTAM have been added to "Appendix A: Supported Terminals." The following summarizes the new 3270 devices that VTAM supports:

Type of Support	Treated as a Non-SNA Terminal	Treated as a SNA Terminal
Supported for Local Attachment	<ul style="list-style-type: none"> <li>3274 Control Unit, Model 1B</li> </ul>	<ul style="list-style-type: none"> <li>3274 Control Unit, Model 1A</li> </ul>
Supported for Remote Attachment	<ul style="list-style-type: none"> <li>3274 Control Unit, Model 1C</li> <li>3276 Control Unit Display Station, Models 1 and 2. (Model 1 operates with VTAM only in 480-character default mode.)</li> </ul>	<ul style="list-style-type: none"> <li>3274 Control Unit, Model 1C</li> <li>3276 Control Unit Display Station, Models 11 and 12. (Model 11 operates with VTAM only in 480-character default mode.)</li> </ul>

The following 3270 devices can be attached to all models of 3274 Control Units:

- 3277 Display Station, Models 1 and 2
- 3278 Display Station, Models 1\* and 2
- 3284 Printer, Models 1 and 2
- 3286 Printer, Models 1 and 2
- 3287 Printer, Models 1\* and 2
- 3288 Line Printer, Model 2
- 3289 Line Printer, Models 1\* and 2

\*Operates with VTAM only in 480-character default mode.

The following 3270 devices can be attached to all models of 3276 Control Unit Display Stations:

- 3278 Display Station, Models 1 and 2
- 3287 Printer, Models 1, 2, and 3

**Addition of HALT CANCEL Command to VTAM under OS/VS2 MVS:** The HALT CANCEL command, which is used to stop VTAM and was formerly available only in VTAM running under OS/VS1, is now available in VTAM running under OS/VS2 MVS.

**DOS/VS Release 34:** Please note that all statements in this manual that refer to DOS/VS Release 33 also apply to DOS/VS Release 34.

**Changed Documentation**

- Figures 3-2, 5-8, and 5-15 have been corrected.
- In Chapter 3, corrections have been made to the list of names that must not be used for major and minor nodes.
- A note has been added to Chapter 7 citing the ability of TCAM application programs running with TCAM Release 10 to communicate directly (without going through VTAM) with terminals attached to communications controllers operating in network control mode.
- "Appendix A: Supported Terminals" has been updated to reflect support of new 3270 devices.
- Other minor technical and editorial corrections have been made.

**Summary of Amendments (October 29, 1976)  
to GC27-6998-3 by TNL GN27-1545**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature**

VTAM Level 2 is described for OS/VS1 Release 6 in which VTAM runs in a user-numbered partition.

**Changed Documentation**

This edition includes a description of configuration restart and the activation process for DOS/VS Release 32 users. Information that applies only to DOS/VS Release 32 is preceded by the phrase *In DOS/VS Release 32*. Information that applies to all systems except DOS/VS Release 32 is preceded by the phrase *In OS/VS and DOS/VS Release 33*.

**Summary of Amendments (June 30, 1976)  
to GC27-6998-2 by Revision GC27-6998-3**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature**

VTAM Level 2 is described for OS/VS2 SVS in addition to DOS/VS, OS/VS1, and OS/VS2 MVS.

**Changed Documentation**

Chapters 1, 2, 3, 4, and 5 have been reorganized and include minor technical changes. Chapters 6, 7, and 8 have editorial and minor technical changes.

**Summary of Amendments (31 May 1977)  
to GC27-6998-3 by TNL GN31-0606**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS  
Announcement of Additional 3270 Devices  
DOS/VS Release 34**

**New Program Features**

**Support of New 3270 Devices:** The new 3270 devices that are supported by VTAM have been added to "Appendix A: Supported Terminals." The following summarizes the new 3270 devices that VTAM supports:

Type of Support	Treated as a Non-SNA Terminal	Treated as a SNA Terminal
Supported for Local Attachment	<ul style="list-style-type: none"> <li>• 3274 Control Unit, Model 1B</li> </ul>	<ul style="list-style-type: none"> <li>• 3274 Control Unit, Model 1A</li> </ul>
Supported for Remote Attachment	<ul style="list-style-type: none"> <li>• 3274 Control Unit, Model 1C</li> <li>• 3276 Control Unit Display Station, Models 1 and 2. (Model 1 operates with VTAM only in 480-character default mode.)</li> </ul>	<ul style="list-style-type: none"> <li>• 3274 Control Unit, Model 1C</li> <li>• 3276 Control Unit Display Station, Models 11 and 12. (Model 11 operates with VTAM only in 480-character default mode.)</li> </ul>

The following 3270 devices can be attached to all models of 3274 Control Units:

- 3277 Display Station, Models 1 and 2
- 3278 Display Station, Models 1\* and 2
- 3284 Printer, Models 1 and 2
- 3286 Printer, Models 1 and 2
- 3287 Printer, Models 1\* and 2
- 3288 Line Printer, Model 2
- 3289 Line Printer, Models 1\* and 2

\*Operates with VTAM only in 480-character default mode.

The following 3270 devices can be attached to all models of 3276 Control Unit Display Stations:

- 3278 Display Station, Models 1 and 2
- 3287 Printer, Models 1, 2, and 3

**Addition of HALT CANCEL Command to VTAM under OS/VS2 MVS:** The HALT CANCEL command, which is used to stop VTAM and was formerly available only in VTAM running under OS/VS1, is now available in VTAM running under OS/VS2 MVS.

**DOS/VS Release 34:** Please note that all statements in this manual that refer to DOS/VS Release 33 also apply to DOS/VS Release 34.

**Changed Documentation**

- Figures 3-2, 5-8, and 5-15 have been corrected.
- In Chapter 3, corrections have been made to the list of names that must not be used for major and minor nodes.
- A note has been added to Chapter 7 citing the ability of TCAM application programs running with TCAM Release 10 to communicate directly (without going through VTAM) with terminals attached to communications controllers operating in network control mode.
- "Appendix A: Supported Terminals" has been updated to reflect support of new 3270 devices.
- Other minor technical and editorial corrections have been made.

**Summary of Amendments (29 October 1976)  
to GC27-6998-3 by TNL GN27-1545**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature** VTAM Level 2 is described for OS/VS1 Release 6 in which VTAM runs in a user-numbered partition.

**Changed Documentation** This edition includes a description of configuration restart and the activation process for DOS/VS Release 32 users. Information that applies only to DOS/VS Release 32 is preceded by the phrase **In DOS/VS Release 32**. Information that applies to all systems except DOS/VS Release 32 is preceded by the phrase **In OS/VS and DOS/VS Release 33**.

**Summary of Amendments (30 June 1976)  
to GC27-6998-2 by Revision GC27-6998-3**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature** VTAM Level 2 is described for OS/VS2 SVS in addition to DOS/VS, OS/VS1, and OS/VS2 MVS.

**Changed Documentation** Chapters 1, 2, 3, 4, and 5 have been reorganized and include minor technical changes. Chapters 6, 7, and 8 have editorial and minor technical changes.

**Summary of Amendments (October 29, 1976)  
to GC27-6998-3 by TNL GN27-1545**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature** VTAM Level 2 is described for OS/VS1 Release 6 in which VTAM runs in a user-numbered partition.

**Changed Documentation** This edition includes a description of configuration restart and the activation process for DOS/VS Release 32 users. Information that applies only to DOS/VS Release 32 is preceded by the phrase **In DOS/VS Release 32**. Information that applies to all systems except DOS/VS Release 32 is preceded by the phrase **In OS/VS and DOS/VS Release 33**.

**Summary of Amendments (June 30, 1976)  
to GC27-6998-2 by Revision GC27-6998-3**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature** VTAM Level 2 is described for OS/VS2 SVS in addition to DOS/VS, OS/VS1, and OS/VS2 MVS.

**Changed Documentation** Chapters 1, 2, 3, 4, and 5 have been reorganized and include minor technical changes. Chapters 6, 7, and 8 have editorial and minor technical changes.





**Summary of Amendments (June 30, 1976)  
to GC27-6998-2 by Revision GC27-6998-3**

**VTAM Level 2 for DOS/VS, OS/VS1, OS/VS2 SVS, and OS/VS2 MVS**

**New Program Feature** VTAM Level 2 is described for OS/VS2 SVS in addition to DOS/VS, OS/VS1, and OS/VS2 MVS.

**Changed Documentation** Chapters 1, 2, 3, 4, and 5 have been reorganized and include minor technical changes. Chapters 6, 7, and 8 have editorial and minor technical changes.

**Summary of Amendments (June 15, 1976)  
to GC27-6998-2 by TNL GN27-1510**

**VTAM Level 2 for DOS/VS, OS/VS1, and OS/VS2 MVS**

**New Program Feature** Support for the IBM 3705-II Communications Controller is described for planning purposes until the 3705-II is available.

**Summary of Amendments (November 20, 1975)  
to GC27-6998-2 by TNL GN27-1497**

**VTAM Level 2 for DOS/VS, OS/VS1, and OS/VS2 MVS**

**New Program Features**

**SENDCMD and RCVCMD Macro Instructions:** A description of the SENDCMD and RCVCMD macro instructions has been added. These macro instructions allow an authorized application program to enter VTAM network operator commands and to receive VTAM network operator messages.

**Configuration Restart Enhancements:** Enhanced configuration restart facilities are described that allow restoration of the VTAM network to its prefailure status after a VTAM or NCP failure.

**Additional Options on START and VARY Commands:** Additional parameters are described on the START VTAM and VARY NET,ACT commands that allow a network operator to activate the VTAM network to its initial status or its status prior to deactivation.

**Manual Switched CPU and NCP Backup:** The ability to manually switch to backup CPU or NCP and restart the network using the WARM option on the START and VARY commands is described.

**Altered TCAM Functions in a Shared Network:** Certain functions supported by TCAM Release 5 that were not supported in a shared VTAM/TCAM network by VTAM Release 1.1 are now supported or replaced with similar VTAM functions.

**Summary of Amendments (August 30, 1975)  
to GC27-6998-1 by Revision GC27-6998-2**

**VTAM Level 2 for DOS/VS and OS/VS**

**New Program Features** **New Terminals:** VTAM Level 2 adds support for certain SNA terminals on switched (dial) SDLC links. VTAM supports the following new terminal products:  
IBM 3270 Information Display System (adapted for SNA)

IBM 3660 Supermarket System  
IBM 3767 Communication Terminal  
IBM 3770 Data Communication System  
IBM System/32 Batch Work Station

**Session Parameters:** VTAM allows application program and terminal-specified session parameters.

**New Logon and Logoff Command:** VTAM allows SNA terminals to send character-coded and field-formatted logon and logoff commands.

## Changed Documentation

**Systems Network Architecture (SNA):** VTAM is described as a component of SNA. The relationship between SNA and VTAM is explained generally in Chapters 1 and 2.

**Reorganization:** Chapter 3, "Creating a VTAM Telecommunication System," has a reorganized description of how to create the VTAM system. Appendix A has been reorganized to reflect the distinction between SNA terminals and local 3270, BSC, and start-stop terminals. A new chapter, Chapter 8, has been added to describe support for local 3270, BSC and start-stop terminals.

**Changed Terminology:** Certain terms previously used by VTAM have been changed to conform to SNA definition of terms. Figure SC-1 summarizes these changes.

This term in the previous edition	Has been changed to this term in this edition
Synchronous flow	Normal flow
Asynchronous flow	Expedited flow
FME response	Definite response 1
RRN response	Definite response 2
Indicator (for session control and data flow control)	Command or reply

Figure SC-1. Terminology Changes in This Edition

## PREFACE

This publication provides an overview of the Virtual Telecommunications Access Method (VTAM). It is directed primarily to data processing managers and system programmers of DOS/VS and OS/VS installations that may install or maintain a telecommunication system that uses VTAM.

The installer of a system that uses VTAM should use this book to get a general understanding of VTAM concepts and the requirements and options that must be considered in planning and installing a VTAM system. Then, to perform specific steps, such as estimating main storage requirements and writing VTAM definition statements, the installer should use the appropriate *VTAM System Programmer's Guide*, the *NCP Generation* publication, relevant operating system publications, and publications that may be required to install the telecommunications terminals and terminal systems that will communicate with VTAM application programs. The programmer who writes VTAM application programs can use this book (although it is not required) to understand the context in which the programs will be executed.

A more general description of VTAM is provided in *Introduction to VTAM*, GC27-6987.

### How This Book Is Organized

- Chapters 1 and 2 provide an introduction to VTAM and describe VTAM's major concepts and facilities.
- Chapters 3, 4, and 5 describe the primary interfaces to VTAM. Chapter 3 describes how to define a VTAM telecommunication system. Chapter 4 describes how to control a VTAM system, and Chapter 5 describes in general how to write VTAM application programs.
- Chapter 6 describes VTAM's reliability, availability, and serviceability features.
- Chapter 7 describes hardware and software requirements for VTAM and discusses planning considerations for functions such as telecommunication security.
- Chapter 8 describes support for local 3270, BSC, and start-stop terminals.

Most of the information in this manual applies for VTAM in all operating systems. Exceptions are shown as follows. For an entire topic, the heading tells which system or systems the information applies to. For example: "Starting VTAM in OS/VS." For a paragraph, a phrase in boldface type tells which system or systems the information applies to. In particular, information that applies only to DOS/VS Release 32 is preceded by the phrase *In DOS/VS Release 32*, and information that applies to all systems except DOS/VS Release 32 is preceded by the phrase *In OS/VS and DOS/VS Release 33*. Minor differences between systems are indicated in the text of the description.

### Related Publications (Other Than VTAM)

Depending on the terminal product with which VTAM is used, the reader may be required to become familiar with certain details of Systems Network Architecture (SNA). Consult the programming publications associated with each SNA terminal product to determine whether this is a requirement. Apart from installation requirements, some readers may wish to become familiar with SNA for general understanding; these readers are referred to *Systems Network Architecture General Information*, GA27-3102.

References are made in this publication to *NCP Generation* publication. The complete title and form number is: *IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual*, GC30-3008.

Other publications (other than VTAM publications) referred to in this publication or related to this publication are listed in the Bibliography at the end of this book.

**Related VTAM Publications**

*VTAM Concepts and Planning* is one of a number of VTAM publications. Figure P-1 shows the reading order of these publications for different user needs.

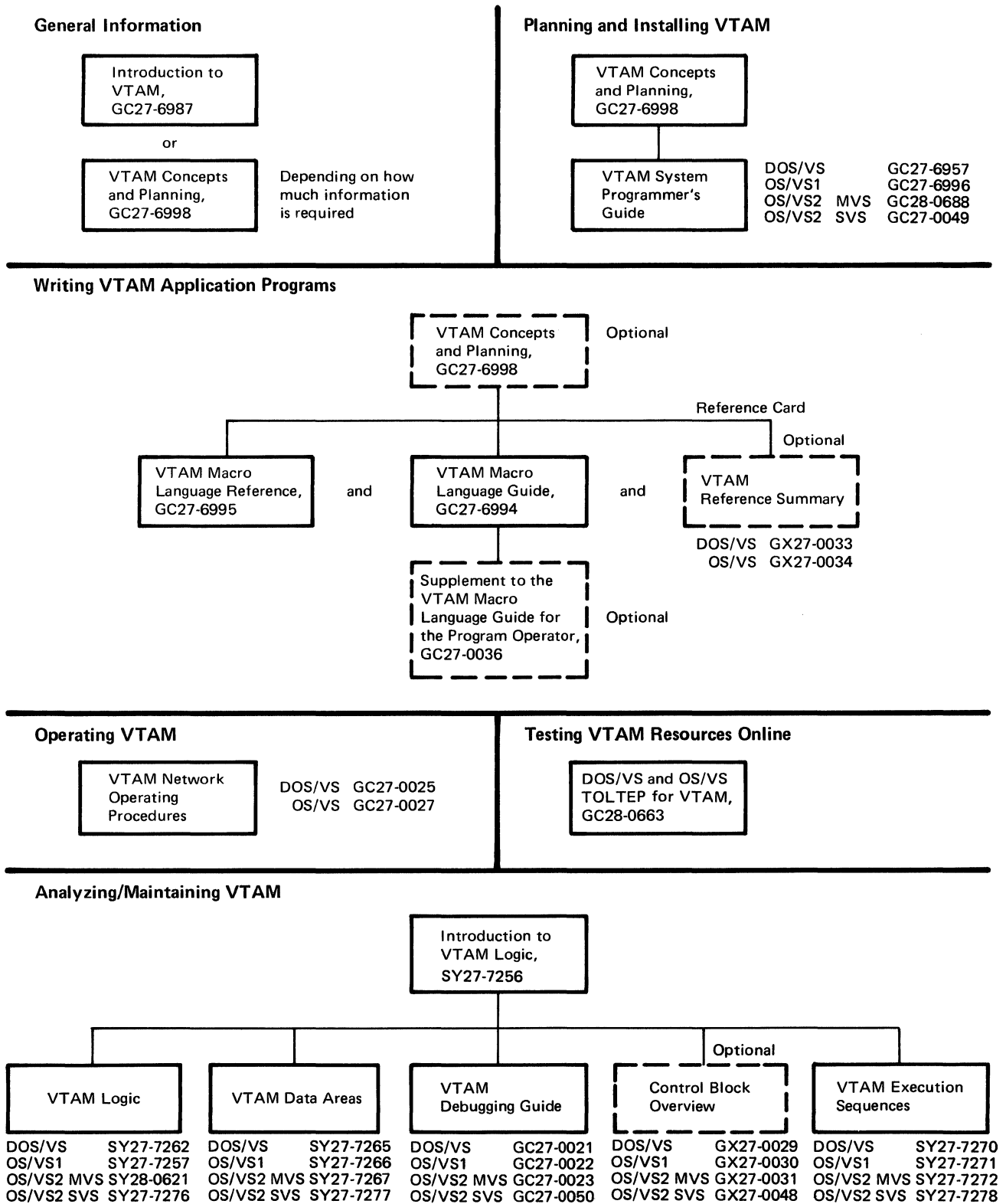


Figure P-1. The VTAM Publications Plan

## Contents

<b>Preface</b> . . . . .	<b>i</b>
<b>Chapter 1. Introduction to VTAM</b> . . . . .	<b>1</b>
Allocation of Network Resources . . . . .	1
Data Flow through a VTAM Telecommunication System . . . . .	1
Sharing Resources . . . . .	2
Distributed Function . . . . .	2
TCAM, BTAM, and QTAM in a VTAM System . . . . .	3
System Requirements for using VTAM . . . . .	3
<b>Chapter 2. A VTAM Telecommunication System</b> . . . . .	<b>5</b>
Elements of a VTAM System . . . . .	5
VTAM Application Programs . . . . .	5
Communications Controllers in a VTAM System . . . . .	5
Terminals in a VTAM System . . . . .	7
SNA Terminals . . . . .	7
Local 3270, BSC, and Start-Stop Terminals . . . . .	8
Four Views of a VTAM System . . . . .	8
Installing a VTAM System . . . . .	9
Creating the VTAM System . . . . .	9
Controlling the VTAM System . . . . .	12
Designing VTAM Application Programs . . . . .	12
Establishing Procedures for Using the System . . . . .	12
VTAM in Operation . . . . .	12
<b>Chapter 3. Creating a VTAM Telecommunication System</b> . . . . .	<b>15</b>
Defining VTAM and Local Devices to the Operating System . . . . .	15
Generating a Network Control Program (NCP) . . . . .	18
Defining the Network to VTAM . . . . .	19
The Major and Minor Node Structure . . . . .	19
Naming Major and Minor Nodes . . . . .	19
Defining Application Program Major Nodes . . . . .	19
Defining Local 3270 Major Nodes . . . . .	21
Defining Local SNA Major Nodes . . . . .	22
Defining NCP Major Nodes . . . . .	22
Defining Switched SNA Major Nodes . . . . .	23
Defining Connection Procedures . . . . .	24
Defining Terminal-Initiated Connection . . . . .	25
Types of Logical Unit Logons . . . . .	25
Defining Field-Formatted Logons . . . . .	26
Defining Character-Coded Logons . . . . .	26
Using the IBM-Supplied Character-Coded (USS) Definition Table . . . . .	26
Defining a Character-Coded (USS) Definition Table . . . . .	27
Defining and Using an Interpret Table . . . . .	28
Defining VTAM-Initiated Connection (Automatic Logon) . . . . .	31
Defining VTAM Application Program-Initiated Connection . . . . .	31
Acquiring Connection (OPNDST with OPTCD=ACQUIRE) . . . . .	31
Simulating a Logon (SIMLOGON) . . . . .	32
Passing Connection to Another Program (CLSDST with OPTCD=PASS) . . . . .	32
Defining Network Operator-Initiated Connection . . . . .	32
Defining Logon Modes . . . . .	32.1
Defining Disconnection Procedures . . . . .	34
Disconnection Requested by the Terminal . . . . .	34
Conditional or Unconditional Disconnection . . . . .	34
Holding the Physical Unit Connection . . . . .	35
Using Request Shutdown Protocol . . . . .	35
Disconnection Requested by the VTAM Application Program . . . . .	35
Disconnection Requested by the Network Operator . . . . .	36
Disconnection Requested by VTAM . . . . .	36
Effect of Disconnection on Automatic Logon . . . . .	36
Authorization, Accounting, and Logon-Interpret Exit Routines . . . . .	36
Authorization Exit Routine . . . . .	37
Accounting Exit Routine . . . . .	38
Logon-Interpret Exit Routine . . . . .	38
Defining VTAM Start Options . . . . .	39
Defining VTAM Buffering . . . . .	40
<b>Chapter 4. Controlling a VTAM System</b> . . . . .	<b>43</b>
Levels of Control . . . . .	43
VTAM Commands . . . . .	43
Starting VTAM . . . . .	43
Starting VTAM in DOS/VS . . . . .	44

Starting VTAM in OS/VS . . . . .	44
Halting VTAM . . . . .	44
Orderly Closedown . . . . .	45
Quick Closedown . . . . .	45
Cancel Closedown . . . . .	45
Designing TPEND for the HALT Command . . . . .	45
Monitoring VTAM Status . . . . .	46
Activating and Deactivating Nodes . . . . .	47
Starting and Stopping an Application Program . . . . .	50
Activating and Deactivating Local 3270s . . . . .	50
Activating and Deactivating Local SNA Major Nodes . . . . .	51
Activating and Deactivating an NCP . . . . .	51
Activating and Deactivating Remote Attachments . . . . .	52
Example of Deactivating and Reactivating a Remotely Attached Logical Unit . . . . .	53
Activating and Deactivating Switched SNA Major Nodes . . . . .	53
Special Considerations for Activation . . . . .	54
Initiating Requests for Connection . . . . .	55
Starting and Stopping VTAM Facilities . . . . .	55
Suppressing Network Operator Messages . . . . .	55
Changing Line-Scheduling Specifications . . . . .	56
Network Operator Control from an Authorized Application Program . . . . .	56
Considerations for Network Operator Control . . . . .	57
<b>Chapter 5. VTAM Application Programs . . . . .</b>	<b>59</b>
The VTAM Language . . . . .	59
VTAM Macro Instructions . . . . .	59
Connection Macro Instructions . . . . .	59
Communication Macro Instructions . . . . .	60
Network Control Macro Instructions . . . . .	60
Control-Block Macro Instructions . . . . .	61
Support Macro Instructions . . . . .	62
Relating VTAM Control Blocks and Executable Macro Instructions . . . . .	62
Opening the Application Program . . . . .	63
Connecting Terminals . . . . .	63
Disconnecting Terminals . . . . .	64
Communicating with Logical Units . . . . .	64
Handling Control Blocks, I/O Areas, and Work Areas . . . . .	64
Organization of the Application Program . . . . .	65
Overlapping VTAM Requests with Other Processing . . . . .	65
EXLST Exit Routines . . . . .	67
Error Notification . . . . .	68
Other Considerations for Program Organization . . . . .	70
Application Program Concepts and Facilities . . . . .	71
Opening the Application Program . . . . .	71
Connection . . . . .	71
Acceptance . . . . .	71
Acquisition . . . . .	72
Establishing Sets of Session Parameters . . . . .	73
Queuing Connection Requests . . . . .	73
Disconnection . . . . .	74
Record-Mode Communication . . . . .	74
Messages and Responses . . . . .	76
Message Flow: Normal and Expedited . . . . .	76
Levels of Message Control: Session Control and Data Flow Control . . . . .	76
Responses: Requesting a Response . . . . .	78
Responses: Positive and Negative . . . . .	78
Responses: Definite Response 1 and 2 . . . . .	78
Sequencing and Chaining . . . . .	78
Sending Messages: Scheduled and Responded Output . . . . .	80
Receiving Input . . . . .	81
Quiesce, Change-Direction and Bracket Protocols . . . . .	84
Request Mode Selection . . . . .	88
Additional Protocols . . . . .	89
Communicating with the 3270 Information Display System in Record Mode . . . . .	89
Basic-Mode and Record-Mode Communication . . . . .	90
Identifying Terminals . . . . .	90
Specific-Mode and Any-Mode . . . . .	90
Continue-Any and Continue-Specific Modes . . . . .	92
Handling Excess Input Data . . . . .	92



<b>Chapter 6. Reliability, Availability, Serviceability</b>	<b>95</b>
VTAM's RAS Strategy	95
VTAM's RAS Facilities	96
Serviceability Aids	96
Error Recording	96
Traces	97
Dumps	98
Teleprocessing Online Test Executive Program (TOLTEP)	99
Reliability and Availability Support	100
Error Detection and Feedback	101
NCP Initial Test	101
Storage Management	101
Error Recovery	102
NCP Slowdown	103
Configuration Restart for OS/VS and DOS/VS Release 33	103
Configuration Restart for DOS/VS Release 32	104
<b>Chapter 7. VTAM Planning Considerations and Requirements</b>	<b>105</b>
Machine Requirements	105
CPU Support	105
Local 3270s	105
Local 3790s	105
Requirements for Communication Controllers	105
Remote Terminals	106
Storage Requirements	107
Operating System Requirements	107
Upward Compatibility	107
Requirements for DOS/VS	107
Requirements for OS/VS	108
Network Control Program Requirements	109
NCP Functions Required by VTAM	109
Defining the NCP to VTAM	110
Pacing	110
Support for SNA Switched Networks	110
Defining a Switched Major Node	111
Generating Switched Line Groups in the NCP	111
ID Verification for Switched SNA Major Nodes	111
Dial-In Operation	112
Dial-Out Operation	112
Disconnecting Physical Units and Logical Units in a Switched Major Node	114
Partitioned Emulation Program (PEP)	114
VTAM Data Sets	115
Data Sets for VTAM under OS/VS	115
Data Requirements for VTAM under DOS/VS	115
Sharing Telecommunication Resources	110
Resources That Can Be Shared	118
Sharing and Managing Resources	119
Managing Resources through VTAM Definition	119
Managing Resources through NCP Generation	120
Managing Resources through Application Programs	121
VTAM Telecommunication Security	121
SNA Identification Verification on Switched Lines	122
Host ID Verification for Physical Units	122
BSC and TWX Identification Verification on Switched Lines	123
Controlling Connections	124
Authorization Exit Routine	124
Acquiring and Passing Connections	124
Controlling Logons	125
Symbolic Names	127
Controlling Access to VTAM	127
Controlling the Use of VTAM Facilities	127
Protecting Confidential Data	128
Other Telecommunication Access Methods	128
TCAM Programs under VTAM	128
TCAM Functions Not Available in a Shared Environment	130
Altered TCAM Operator Control Functions	131
TCAM Operator Control Functions Replaced by VTAM Operator Control Functions	131
TCAM Operator Messages Replaced with VTAM Operator Messages	132
Other Telecommunication Access Methods in DOS/VS	132
Other Telecommunication Access Methods in OS/VS	133

<b>Chapter 8. Support for Local 3270, BSC, and Start-Stop Terminals</b>	<b>135</b>
Using BTAM	135
Using VTAM	135
Topics Not Applicable to Local 3270, BSC, and Start-Stop Terminals	139
<b>Creating a VTAM System That Includes Local 3270, BSC, or Start-Stop Terminals</b>	<b>139</b>
The Network Solicitor	140
How the Network Solicitor Works	140
Modifying the Network Solicitor	141
Replacing the Network Solicitor	142
Specifying Interpret Tables	143
<b>Defining Terminal-Initiated Logons for Local 3270, BSC, and Start-Stop Terminals</b>	<b>144</b>
<b>Defining a Switched Network for Start-Stop and BSC Terminals</b>	<b>145</b>
Call-In Terminals	146
Call-Out Terminals	147
Call-In/Call-Out Terminals	147
<b>Operating a VTAM System with Local 3270, BSC, and Start-Stop Terminals</b>	<b>147</b>
<b>Writing a VTAM Application Program</b>	<b>148</b>
Basic-Mode Concepts	148
The Basic-Mode Macro Instructions	148
The LDO Control Block	149
<b>Communicating with Local 3270, BSC, and Start-Stop Terminals</b>	<b>149</b>
Solicitation	149
Special I/O Operations	151
Special Processing Options	152
Communicating with a BSC or Local 3270 Terminal in Record Mode	152
<b>Appendix A. Supported Terminals</b>	<b>155</b>
SNA Terminals	155
Local SNA Terminals	155
Remote SNA Terminals	156
Non-SNA Terminals	158
Local Non-SNA Terminals	159
Start-Stop Terminals (Remote)	159
Binary Synchronous Communications (BSC) Terminals (Remote)	162
<b>Appendix B. Remote Station Versus Remote Controller</b>	<b>169</b>
<b>Appendix C. Summary of Message Control Information</b>	<b>173</b>
<b>Appendix D. Compatibility Considerations for BSC and Local 3270 Terminals and SNA 3270 Terminals</b>	<b>177</b>
<b>Glossary</b>	<b>179</b>
<b>Bibliography</b>	<b>185</b>
<b>Index</b>	<b>187</b>

## Figures

Figure 1-1.	Data Flow through a VTAM Telecommunication System . . . . .	2
Figure 2-1.	The Major Elements of a VTAM Telecommunication System . . . . .	6
Figure 2-2.	Four Views of a VTAM Telecommunication System . . . . .	10
Figure 3-1.	The Steps in Creating a VTAM Telecommunication System . . . . .	16
Figure 3-2.	Major and Minor Nodes in a VTAM Telecommunication System . . . . .	20
Figure 3-3.	Generating NCP Support in a VTAM Telecommunication System . . . . .	24
Figure 3-4.	The Form of Logon/Logoff Required or More Commonly Used by Several SNA Terminals . . . . .	26
Figure 3-5.	A Field-Formatted Logon . . . . .	27
Figure 3-6.	A Character-Coded Logon . . . . .	28
Figure 3-7.	Defining a USS Definition Table . . . . .	29
Figure 3-8.	Example of VTAM Handling a Character-Coded (USS) Logon . . . . .	30
Figure 3-9.	Defining a Logon Mode Table . . . . .	33
Figure 3-10.	How VTAM Buffers Input and Output Data . . . . .	41
Figure 4-1.	Network Operator Control of VTAM System . . . . .	58
Figure 5-1.	Processing Pattern for a Synchronous Request . . . . .	65
Figure 5-2.	Processing Pattern When an ECB Is Used with an Asynchronous Request . . . . .	66
Figure 5-3.	Processing Pattern When an RPL Exit Routine Is Used with an Asynchronous Request . . . . .	67
Figure 5-4.	A Possible Processing Pattern When Asynchronous Requests Are Issued in RPL Exit Routines . . . . .	68
Figure 5-5.	Processing Pattern for Reporting Errors during an Asynchronous Operation . . . . .	69
Figure 5-6.	Queued and Unqueued Connection Requests . . . . .	75
Figure 5-7.	An Example of Start-Data-Traffic and Clear Commands . . . . .	77
Figure 5-8.	An Example of a Logical Unit Requesting (A) A Definite Response and (B) Only an Exception Response . . . . .	79
Figure 5-9.	An Example of Message Chaining . . . . .	81
Figure 5-10.	Scheduled Output . . . . .	82
Figure 5-11.	Responded Output . . . . .	83
Figure 5-12.	Types of Information Exchanged Between an Application Program and Terminals . . . . .	84
Figure 5-13.	An Example of Quiesce Protocol . . . . .	85
Figure 5-14.	An Example of Change-Direction Protocol . . . . .	86
Figure 5-15.	An Example of Bracket Protocol . . . . .	87
Figure 5-16.	Commands and Indicators Used to Direct the Flow of Data . . . . .	88
Figure 5-17.	Using a Combination of Any-Mode and Specific-Mode to Obtain Data . . . . .	91
Figure 5-18.	Using the Continue-Any and Continue-Specific Modes to Handle Concurrent Inquiries . . . . .	92
Figure 7-1.	Dial-In Operation . . . . .	112
Figure 7-2.	A Path for a Dial-Out Operation . . . . .	113
Figure 7-3.	An Alternate Path for a Dial-Out Operation with an Automatic Calling Unit . . . . .	114
Figure 7-4.	Table of Data Sets Used by VTAM under OS/VS . . . . .	116
Figure 7-5.	Table of Files Used by VTAM under DOS/VS . . . . .	118
Figure 7-6.	VTAM's Use of Libraries under DOS/VS . . . . .	119
Figure 7-7.	Communications Controllers and Transmission Control Units in a Telecommunication Network . . . . .	129
Figure 7-8.	Other Telecommunication Access Methods in DOS/VS . . . . .	133
Figure 7-9.	Other Telecommunication Access Methods in OS/VS . . . . .	134
Figure 8-1.	Major Similarities and Differences Between VTAM and BTAM Application Programs . . . . .	136
Figure 8-2.	Using BTAM and VTAM to Communicate with Local 3270, BSC, and Start-Stop Terminals . . . . .	138
Figure 8-3.	Processing a Terminal-Initiated Logon with the Network Solicitor . . . . .	141
Figure 8-4.	Filing Interpret Tables . . . . .	143
Figure 8-5.	Providing Control Information for Processing Logon Requests from Local 3270, BSC, and Start-Stop Terminals . . . . .	145
Figure 8-6.	Implicit and Explicit Solicitation Using Basic Mode . . . . .	150
Figure B-1.	A Remotely Attached Communications Controller . . . . .	169
Figure B-2.	Communications Controllers Attached as Part of Remote Stations . . . . .	170

## CHAPTER 1. INTRODUCTION TO VTAM

The IBM Virtual Telecommunications Access Method (VTAM) directs the transmission of data between application programs and terminals in a telecommunications network. Application programs using VTAM communicate with terminals without concern for intermediate connections such as communications controllers and communication lines. To support communication within the network, VTAM:

- Controls the allocation of network resources

- Establishes, terminates, and controls connections between application programs and terminals

- Transfers data between application programs and terminals

- Permits application programs to share resources such as communication lines, communications controllers, and terminals

- Permits communications controllers and terminals to perform some network functions

- Permits concurrent execution of TCAM and VTAM application programs using the same network (except in OS/VS2 SVS)

- Permits the operation of the network to be monitored

- Permits the configuration of the network to be changed while the network is being used

### Allocation of Network Resources

VTAM controls the allocation of all network resources; that is, VTAM controls who uses what resources. The only aspect of resource allocation of direct concern to an application program is the issuing of requests to have terminals connected to it. VTAM allocates intermediate elements only for the time needed to satisfy a specific transmission request. By owning and controlling all resources, VTAM provides a focal point within the system for controlling the network.

### Data Flow through a VTAM Telecommunication System

VTAM manages the flow of data between itself and the application program; between the host computer (using the I/O facilities of the operating system) and the communications controller; and between the communications controller (using the facilities of the NCP) and the terminal.

VTAM is responsible for the transfer of data between the elements in only the VTAM portion of a telecommunication system. VTAM application programs and terminals mark the limits of the VTAM system. Control of the data flow beyond these limits is the responsibility of elements at those limits.

In a telecommunications system with SNA terminal systems or System/370s used as remote stations, some communication occurs beyond VTAM's control. Figure 1-1 shows the data flow for a VTAM system that has an SNA terminal system. Note that VTAM manages only part of the data transfer between the application program and the final destination, the work station. The application program in the SNA terminal system is responsible for the transfer of the data between the logical unit and the work station. (Logical units are described in Chapter 2.)

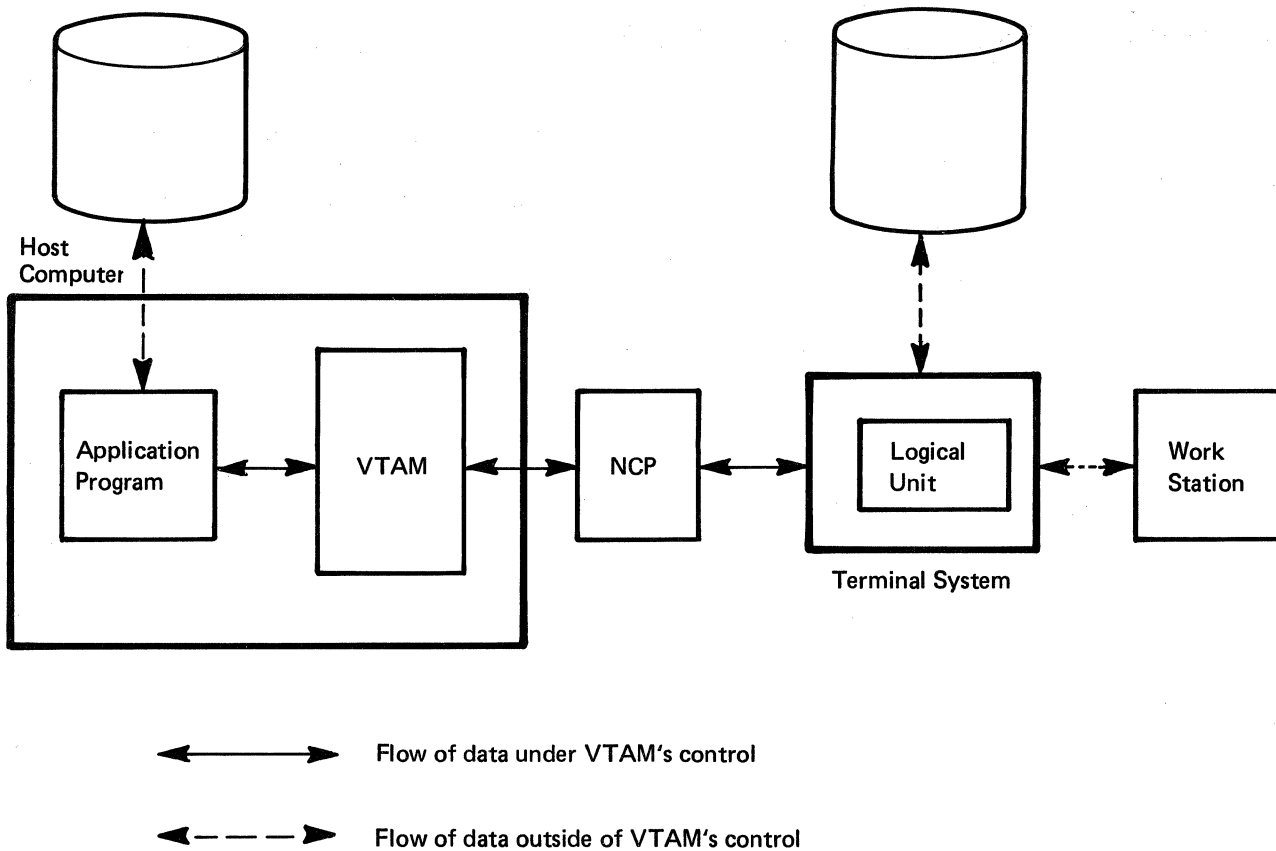


Figure 1-1. Data Flow through a VTAM Telecommunication System

VTAM application programs are responsible for transferring data using auxiliary storage devices. The application program can use an access method such as IBM's Virtual Storage Access Method (VSAM) for access to auxiliary storage of the host system.

## Sharing Resources

VTAM allows resources to be shared among users of the network. An application program can communicate with several terminals simultaneously and any one terminal may communicate with any application program using VTAM. However, once a terminal is connected to an application program, that terminal can communicate only with that application program until released by the program.

Resources that make up the paths between application programs and terminals are also shared. VTAM uses path elements (such as communications controllers and lines) on behalf of an application program and terminal only as long as needed to complete a specific data transfer request. For example, two terminals can be attached on the same multipoint line, and each terminal can be connected to a different application program. When either application program requests data transfer, the multipoint line is used. Thus, the line is shared among the terminals and application programs.

## Distributed Function

In a VTAM system, network control is distributed throughout the network rather than concentrated in the host computer. Communication controllers control line scheduling

and polling. Programmable terminals control the network beyond VTAM's control. Because these functions are distributed, data transfer and data processing can occur simultaneously, and more of the host's capacity can be applied to data processing.

## **TCAM, BTAM, and QTAM in a VTAM System**

In an OS/VS system, BTAM, TCAM, and VTAM can operate concurrently. Application programs can use TCAM through VTAM; that is, application programs written for TCAM can be executed using VTAM facilities and resources. BTAM programs can be executed in a system with VTAM, but they must use a separate network.

In a DOS/VS system, QTAM, BTAM, and VTAM can operate concurrently. QTAM and BTAM programs do not interact with VTAM.

## **System Requirements for Using VTAM**

VTAM can be a component of the DOS/VS, the OS/VS1, and the OS/VS2 operating systems. The DOS/VS supervisor must be generated to include multiprogramming support.

The System/370 instruction set must include the Compare and Swap and the Compare Double and Swap instructions. These instructions are part of a hardware feature available on System/370 CPUs.

VTAM requires a 3704 or 3705 Communications Controller in network control mode to support remote communications controllers and terminals. Appendix A lists terminals that can be used with VTAM.

Chapter 7 describes system requirements for using VTAM in more detail.



## CHAPTER 2. A VTAM TELECOMMUNICATION SYSTEM

VTAM coordinates the activities of the elements that make up a VTAM system. This chapter describes these elements, the relationship of the elements, and the steps involved in creating a telecommunication system using these elements.

### Elements of a VTAM System

Figure 2-1 shows the major elements of a VTAM system. Each element is described below.

#### ***VTAM Application Programs***

A VTAM application program is any program that uses VTAM macro instructions. A VTAM application program is independent of line activities (such as line scheduling) and attachment concerns (such as whether a terminal is locally or remotely attached); is capable of referring to terminals symbolically; and communicates with terminals in real-time basis.

Before using any VTAM facilities, an application program must identify itself and be connected to VTAM by opening an access method control block (ACB). This control block represents the application program to VTAM and indicates the definition statement that describes the program's characteristics, such as authorization to use certain facilities.

A VTAM application program can use any of the facilities available to other programs in the host computer. A program can perform a single function or many. VTAM permits the part of a VTAM application program that processes data to be separate from the part that communicates with terminals. This separation allows each part to be created separately and means that changes or additions to one part need not affect other parts. The processing part of an application program may be written in a higher level language, such as PL/I; the telecommunication part, using VTAM macro instructions, must be written in assembler language. When using programmable terminals, the user decides which processing is performed by the processing part of the VTAM application program and which is performed by an application program in the terminal.

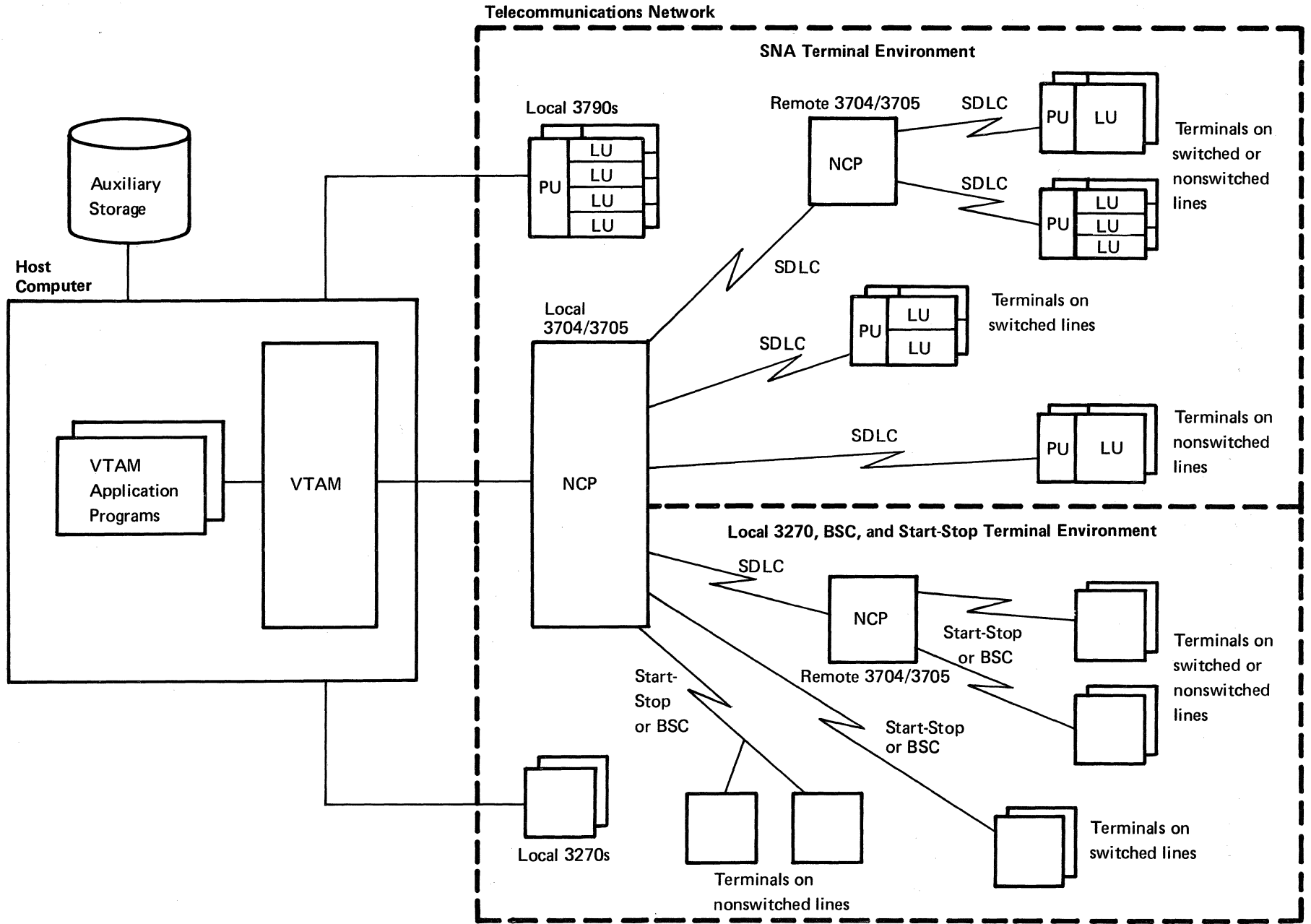
#### ***Communications Controllers in a VTAM System***

VTAM uses the IBM 3704 and 3705 Communications Controllers to communicate with the remote network. These controllers can be either *locally attached* (that is, connected to the host computer by a data channel) or *remotely attached* (that is, connected to a locally attached communications controller by a communication line). Communications controllers link VTAM with the remote portions of the network and control the flow of information between terminals and VTAM.

The communications controllers are programmable devices; the program that controls these devices for VTAM is called the network control program/VS (referred to as *NCP* in this publication). The NCP has generation options that allow a communications controller to be operated in different modes. When operated in *network control mode*, the communications controller allows its network resources to be shared. Communications controllers can also be operated in *emulation mode*; when operated in this mode, they emulate the IBM 2701 Data Adapter Unit and the IBM 2702 and 2703 Transmission Control Units. In addition, an NCP can be generated with partitioned emulation programming (PEP), which allows a communications controller to handle separate telecommunication lines in either network control or emulation mode at the same time. VTAM uses only the network control mode of the NCP, with or without PEP.



Figure 2-1. The Major Elements in a VTAM Telecommunications System



The NCP allows some functions previously performed entirely in the host computer to be performed in the communications controller. Among the functions now provided by the communications controller are:

- Controlling lines
- Controlling dynamic buffering
- Deleting and inserting line control characters
- Detecting permanent line errors
- Gathering line statistics
- Activating and deactivating lines
- Closing down the network
- Handling recoverable errors
- Providing error statistics to VTAM

In addition, for start-stop and BSC terminals:

- Translating character codes
- Stamping dates and times

By performing these functions outside the host computer and by allowing much of the network traffic to be controlled in the network, the NCP conserves host computing resources.

Although these activities are actually performed in the communications controllers, they are controlled by VTAM. For example, when the NCP is to deactivate a line, the command to deactivate comes from VTAM. VTAM, in turn, is controlled by the user's definition of the system, by the network operator, and by the VTAM applications programs.

### ***Terminals in a VTAM System***

VTAM supports certain remote terminals (that is, terminals connected to a communications controller by a communication line) that use the following line disciplines:

- Synchronous data link control (SDLC)
- Start-stop
- Binary synchronous communications (BSC)

The type of line discipline used depends upon the terminal. Appendix A lists remote devices supported by VTAM for each line discipline.

In addition to supporting remote terminals, VTAM supports the local attachment of 3270 terminals and 3790 terminal systems through a data channel to the host computer.

### **SNA Terminals**

Each SNA terminal is supported remotely using SDLC switched or nonswitched lines or locally (through a channel interface). To VTAM, an SNA terminal is either a physical unit (PU) or a logical unit (LU) associated with a physical unit. The physical units and logical units perform functions defined by SNA. In general, a physical unit corresponds to a controller or control unit that is attached to a line or channel; a logical unit corresponds to an addressable unit of logic within that controller or control unit. The relationship of the logical unit with the other components in an SNA terminal product depends upon the particular product. See the programming publication for the terminal product for a description of that relationship.

To a VTAM application program, an SNA terminal is a logical unit; the program is not aware of the physical unit with which a logical unit is associated. Before a VTAM application program begins to communicate with a logical unit, VTAM must communicate with the associated physical unit as well as with the logical unit.

### **Local 3270, BSC, and Start-Stop Terminals**

VTAM supports remote BSC and start-stop terminals (such as the IBM 3780 Data Communications Terminal and the IBM 2741 Communication Terminal) and local 3270 Information Display System terminals (displays or printers attached to a 3272 Control Unit that is attached to a host computer by a data channel). With these devices, there is a one-to-one relationship between the VTAM terminals and the physical devices; for example, a 2741 is a terminal to VTAM and also a physical end of a network.

## **Four Views of a VTAM System**

To understand VTAM, a user should know what the system looks like from four viewpoints. These viewpoints are shown in Figure 2-2, which depicts the system: (A) as seen in terms of its physical components, (B) as seen by the operating system, (C) as seen by VTAM, and (D) as seen by application programs.

Part A of Figure 2-2 shows a possible physical configuration of the system. A VTAM system includes a host computer with a system console. The system console is used to enter VTAM network operator commands to control the telecommunication system. Also attached to the host computer are auxiliary storage devices that contain data sets used by VTAM.

The network shown in Section A includes a local 3270 Information Display System, a local 3790 Communication System, and a local 3705 Communication Controller. Attached to the local communications controller is a 3771 Communication Terminal, a 3600 Finance Communication System, and a remote 3705 Communications Controller. Attached to the remote communications controller is a 2741 Communication Terminal and a 3767 Communication Terminal.

A telecommunication system has a definite physical configuration, but its definition and uses are different for the operating system, VTAM, and application programs. Parts B, C, and D of Figure 2-2 show these differences.

Part B of Figure 2-2 depicts the VTAM system as viewed by the operating system. Support is generated in the operating system only for the system console, the auxiliary storage devices, and the local devices of the network. Support for remote devices need not be generated during system generation if they are to be used only through VTAM; such support is generated within VTAM.

The number of auxiliary storage devices used by VTAM depends upon data requirements which in turn are influenced by factors such as the size and complexity of the telecommunication system. In general, data used by or generated by VTAM falls into one of three categories as shown in Part B of Figure 2-2.

*VTAM libraries*, which contain VTAM load modules, descriptions of the telecommunication system, and operational specifications of the installation

*NCP libraries*, which contain NCP load modules and dump records

*RAS (reliability, availability, and serviceability) libraries*, which contain records to assist in error recording and maintenance of the VTAM system

VTAM and NCP libraries include VTAM, NCP, and operating system data sets; most of the library requirements for RAS involve operating system data sets. The composition and organization of these libraries depend upon the operating system under which VTAM is being executed. (See Chapter 7 for details on operating system requirements and on data set requirements for VTAM.)

Part C of Figure 2-2 depicts the telecommunication system as viewed by VTAM. The host computer must contain the operating system (DOS/VS, OS/VS1, or OS/VS2), VTAM, and one or more application programs. To VTAM, an active application program is an open (that is, an initialized) access method control block (ACB). As shown in Part B, all local devices are initially “owned” by the operating system, but, as indicated in Part C, when VTAM is started and begins activating parts of the telecommunication network, VTAM acquires the use of these devices.

VTAM sees each SNA terminal product as a physical unit (PU) and one or more logical units (LUs). Note that the number of logical units in an SNA terminal system is independent of the number of attached devices. VTAM sees each non-SNA terminal product as a single terminal or as a cluster controller with an associated terminal.

The system console is used by VTAM but is not allocated to VTAM. The network operator enters VTAM commands through this console, and VTAM transmits messages to the network operator at this console. (An application program can also enter VTAM operator commands. This facility is described in Chapter 4.)

Part D of Figure 2-2 shows the telecommunication system as viewed by the application program. This view results from VTAM’s ownership of all elements in the network and the way VTAM allocates them. VTAM connects application programs to logical units and non-SNA terminals (local 3270, BSC, and start-stop terminals)<sup>1</sup>. The intermediate elements are allocated only for the time needed to satisfy a specific transmission request.

Application programs are also not directly concerned with the system console used by the network operator or with the VTAM, NCP, or RAS libraries.

## Installing a VTAM System

To install a VTAM system, it must be created, procedures should be defined for using it, the active system must be controlled, and application programs must be designed and coded for the host computer. These steps are discussed below.

### *Creating the VTAM System*

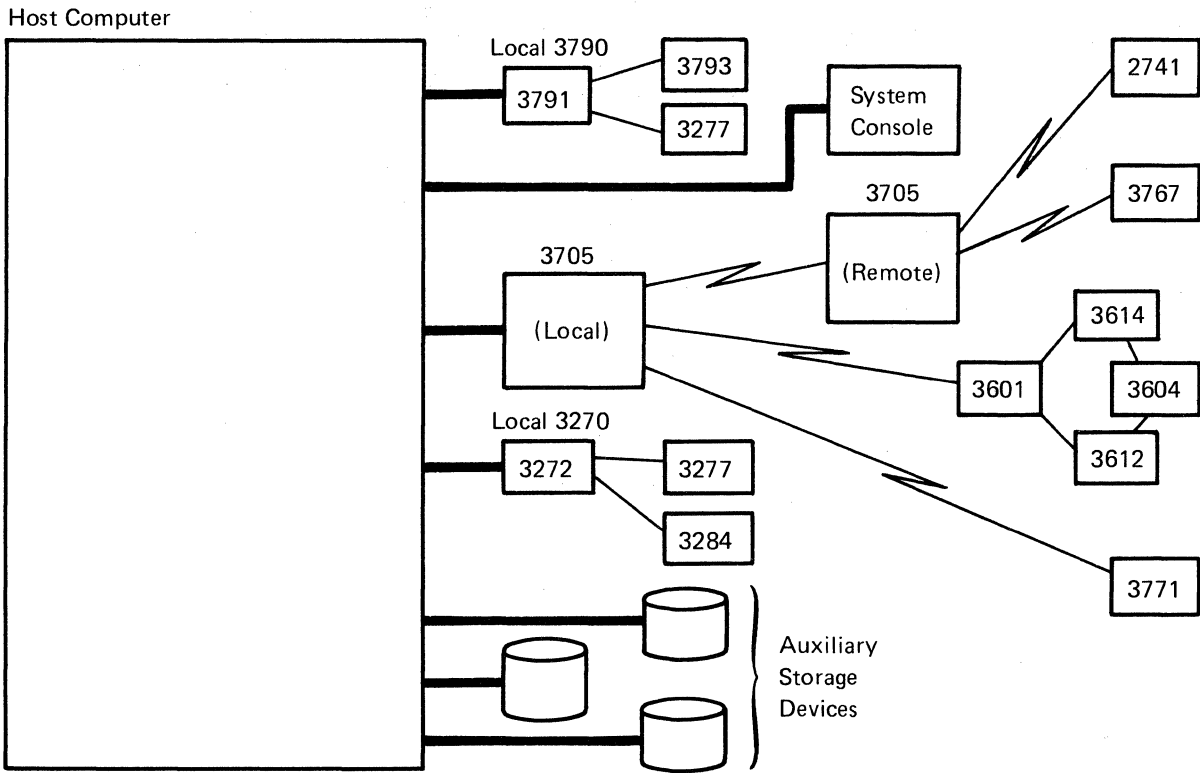
A VTAM telecommunication system is created by generating VTAM into the operating system, defining the network to the communications controllers and to VTAM, and defining special VTAM facilities.

Generating VTAM is part of operating system generation. Defining the network to VTAM is a separate process of identifying and describing elements of the network and then filing these definitions in a VTAM library. Defining special VTAM facilities includes coding exit routines that perform functions such as checking the validity of connection requests between application programs and terminals, collecting accounting information, and structuring VTAM’s logon facility to the user’s specifications. Chapter 3 describes these procedures in detail.

---

<sup>1</sup>In this publication, logical units and non-SNA terminals are both called *terminals* unless a distinction is necessary.

**A PHYSICAL CONFIGURATION OF A TELECOMMUNICATION SYSTEM**



**C THE TELECOMMUNICATION SYSTEM VIEWED BY VTAM**

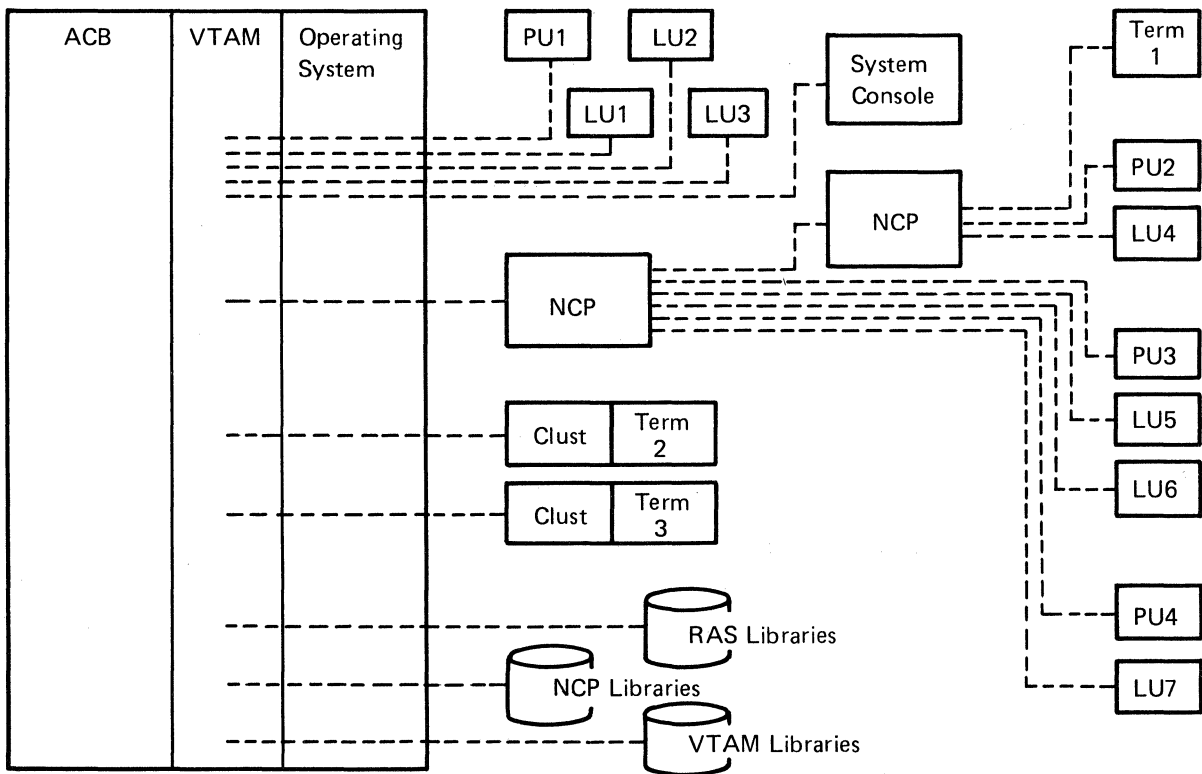
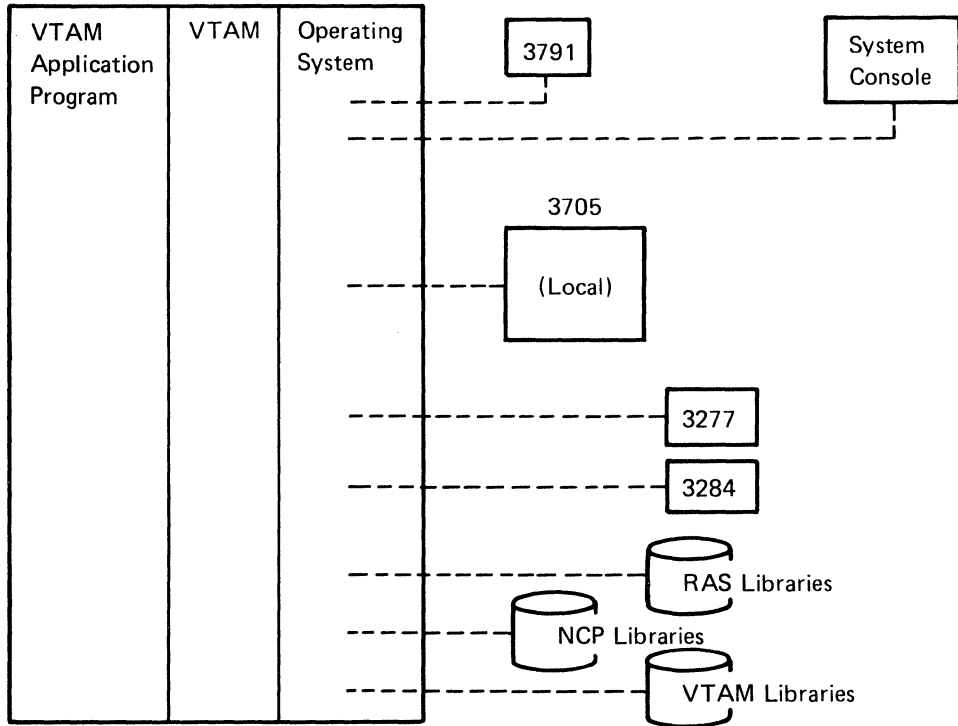


Figure 2-2 (Part 1 of 2). Four Views of a VTAM Telecommunication System

**B THE TELECOMMUNICATION SYSTEM VIEWED BY THE OPERATING SYSTEM**



**D THE TELECOMMUNICATION SYSTEM VIEWED BY VTAM APPLICATION PROGRAMS**

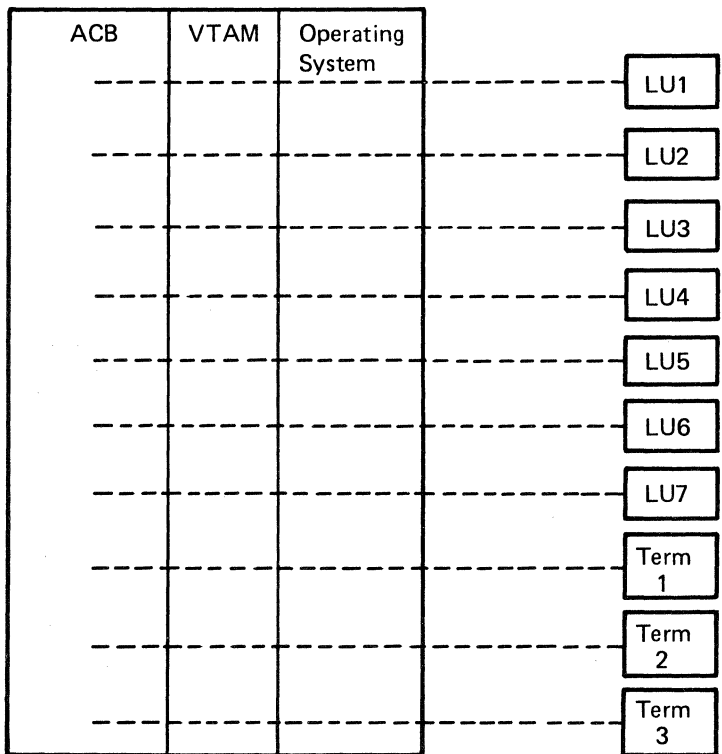


Figure 2-2 (Part 2 of 2). Four Views of a VTAM Telecommunication System

### ***Controlling the VTAM System***

VTAM enables a network operator to dynamically control the VTAM system. The network operator can start and stop VTAM, monitor the activity of the telecommunication system, activate and deactivate network elements such as data links, physical units, and logical units, and start and stop specified VTAM facilities. To perform these functions, the network operator uses a set of VTAM commands. See Chapter 4 for detailed information on the responsibilities and actions of the network operator and for a description of the VTAM network operator facilities.

### ***Designing VTAM Application Programs***

VTAM enables application programs to request connection with specific terminals and to request the transfer of data between the application programs and their connected terminals. The application program can request most VTAM services synchronously (the program waits while VTAM processes the request) or asynchronously (the program continues execution and is interrupted when VTAM has processed the request). Chapter 5 introduces the VTAM facilities available to the application program.

### ***Establishing Procedures for Using the System***

Once a VTAM telecommunication system has been started, it is available to application programs, terminal operators, and the network operator. To ensure that the telecommunication system is used effectively and efficiently, procedures for users of the system and controls that monitor these procedures must be established.

Procedures should be established for the network operator for starting, stopping, and modifying the VTAM system. The user must define how and when to activate and deactivate nodes and specific VTAM functions. The user must also define what to do when error conditions are encountered and what action is to be taken to avoid unnecessary downtime. These actions might include responding to error messages, collecting status information, or correcting the problem.

The application programmer needs to know the conventions to be followed when connecting programs to VTAM and to terminals. Procedures should also be established for the interaction between the application program and the rest of the system. Such procedures might include passing terminal connections between application programs and reacting to system closedown.

The terminal operator needs to know how to log on to and log off from application programs.

Controls should be established to ensure that only authorized users can gain access to VTAM resources. VTAM facilities can be used to control connections between application programs and terminals. Facilities are also available to restrict the use of certain VTAM services to authorized users and to protect confidential data.

Chapter 6 discusses the reliability, availability, and serviceability (RAS) capabilities of VTAM. Chapter 7 describes various VTAM planning considerations.

## **VTAM in Operation**

When VTAM is started, it initiates the telecommunication system according to the specifications of the user. Once VTAM has been started, active application programs can be connected with active terminals in the telecommunication network. As long as a terminal is connected to an application program, it can communicate with that application program.

Before being connected to a terminal, an application program must identify itself to VTAM. Connection can be initiated by a terminal, an application program, the network

operator, or VTAM. When connection is initiated from outside the application program (or the application program simulates an external connection request), the terminal is queued, and the application program is notified of the request. The application program must accept the terminal to complete the connection. The application program can also acquire a terminal directly. Connection is made to the terminal, not the line. When the connection request is completed, the application program is able to transmit data to the terminal by means of input/output requests.

In transmitting data to a terminal, the data is moved from the application program's output data areas to VTAM buffers. VTAM then transmits the data to the terminal (through the NCP for remote devices). Input from the terminals travels the same (but reverse) route. The transmission moves from the terminal to VTAM (through the NCP for remote devices). VTAM then moves the information to the application program's input areas.

When an application program no longer needs a terminal, it can disconnect the terminal. VTAM can then reallocate the terminal to another application program.

When the telecommunication system is to be closed down, VTAM allows the user to terminate processing in an orderly manner and to cease telecommunication activity. From the time that VTAM is started to the time it is terminated, the network operator controls and monitors the telecommunication system. Most modifications to the network can be made dynamically, without terminating VTAM.





## CHAPTER 3. CREATING A VTAM TELECOMMUNICATION SYSTEM

Creating a VTAM telecommunication system consists of:

- Defining VTAM and local devices to the operating system

- Generating a network control program (NCP)

- Defining the network and application programs to VTAM

- Defining connection and disconnection procedures for terminals in the network

- Coding and including accounting, authorization, and logon-interpret exit routines

- Defining start options (the status of the system when it is initialized and the size, thresholds, and other characteristics of VTAM storage pools to be used for incoming and outgoing data and for control blocks)

This chapter describes in general how a VTAM telecommunication system is created to meet installation requirements. The *(VTAM System Programmer's Guide* for each operating system describes these procedures in detail). Creating a VTAM telecommunication system may also include defining network operating procedures, described in Chapter 4, and writing VTAM application programs, described in Chapter 5.

Figure 3-1 summarizes the steps in creating a VTAM telecommunication system that are discussed in this chapter. Some of the steps in Figure 3-1 are related to other steps. For example, step C, defining the network to VTAM, consists not only of describing the structure of the network to VTAM but also of relating each described logical unit to an associated logon mode table. Thus, step C also involves step E, defining connection and disconnection procedures. The steps in Figure 3-1, based on the separate sets of statements that may be required in setting up the system, are the basis for the organization of this chapter and are a convenient way to look at the processes involved in planning and setting up a VTAM system.

### Defining VTAM and Local Devices to the Operating System

During system generation, VTAM modules are generated and included in the operating system. To include the VTAM modules and the required support in the operating system, the following are specified in the input stream for the first stage of system generation:

- VTAM is specified as a parameter of the TP operand of the SUPVR macro instruction for DOS/VS, or VTAM is specified as a parameter of the ACSMETH operand of the DATAMGT macro instruction for OS/VS.

- Control unit and device statements are specified only for locally attached devices that use VTAM, that is, for local 3270s, local 3790s, and local communications controllers.

Remote devices are not specified during system generation. These devices are specified and described during network definition. Remember that network definition can be done without disrupting other jobs in the operating system. See "Defining the Network to VTAM," later in this chapter, for a description of network definition.

Additional operating system support for VTAM can be included at system generation. See "Operating System Requirements" in Chapter 7 for details on operating system support.

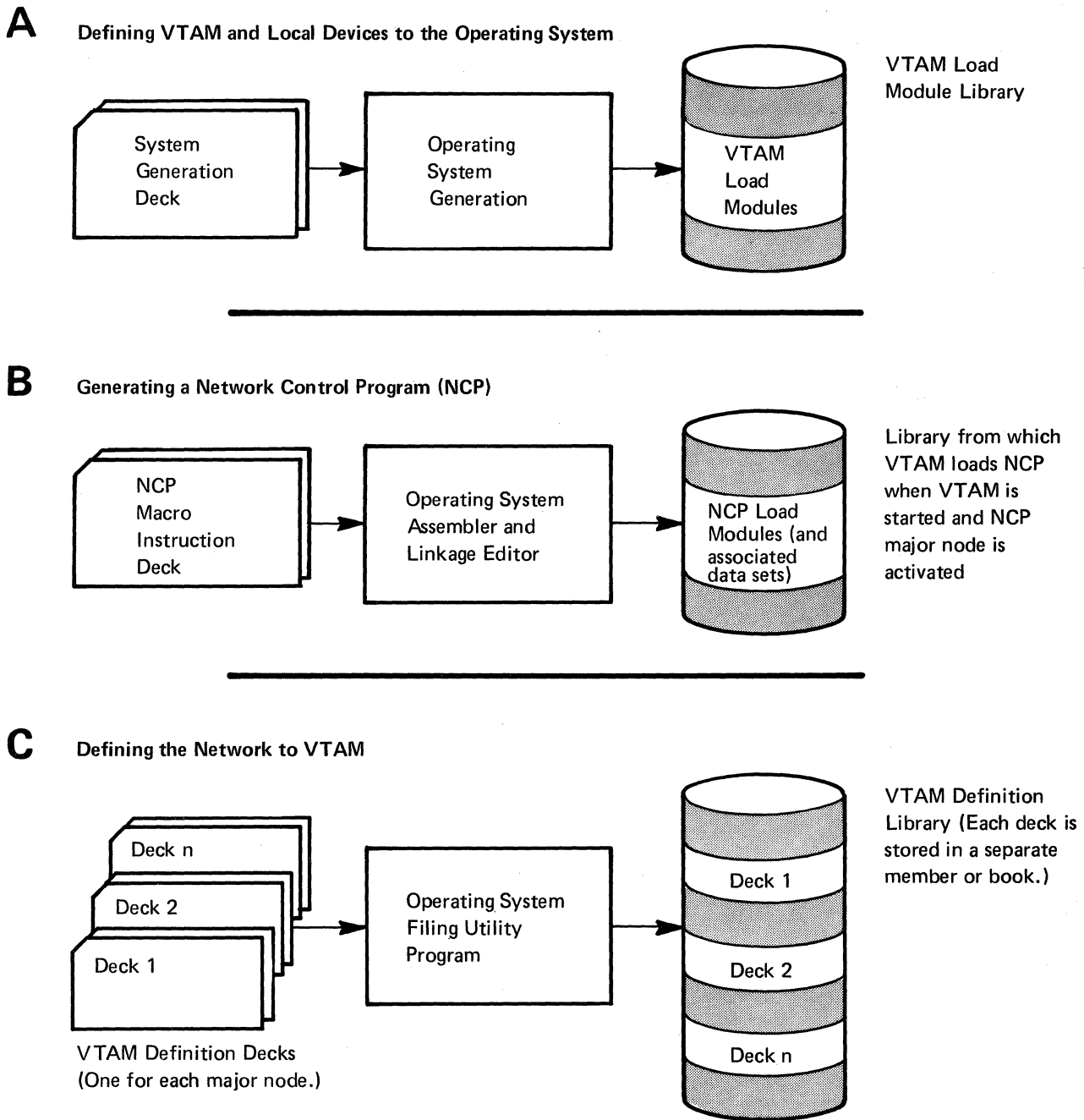
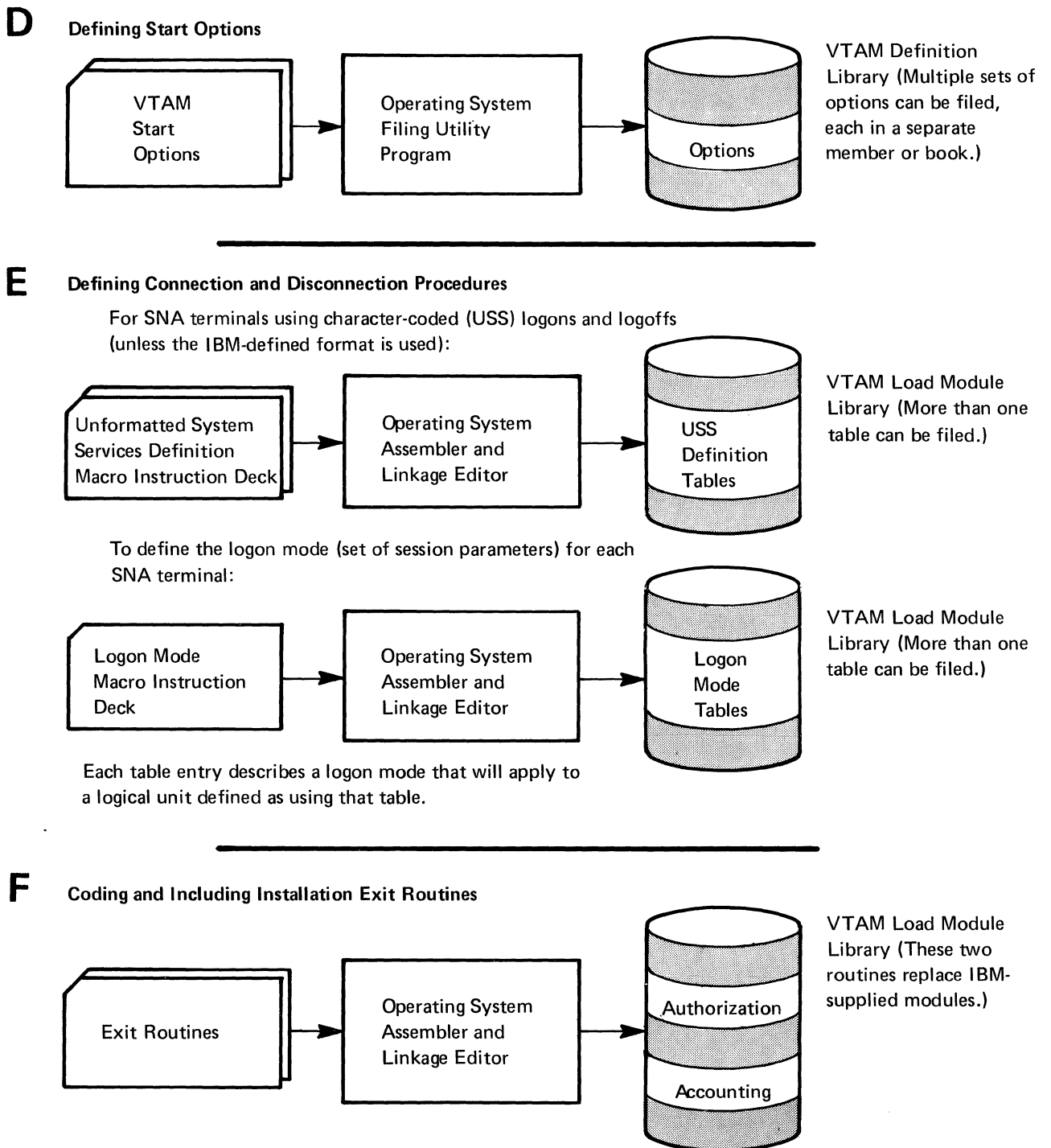


Figure 3-1 (Part 1 of 2). The Steps in Creating a VTAM Telecommunications System



Note: See Figure 7-4 or 7-5 for a summary of data sets or files that are created.

Figure 3-1 (Part 2 of 2). The Steps in Creating a VTAM Telecommunications System

## Generating a Network Control Program (NCP)

If the VTAM telecommunication system includes remote terminals, they must be connected by communication lines to a 3704 or 3705 Communications Controller that contains a network control program (NCP). In creating this system, an NCP must be coded using NCP macro instructions. The macro instructions are assembled, and the object program is then filed so that it can be loaded into the communications controller when the VTAM system is started. (The NCP is loaded automatically (if required) as the result of starting VTAM.) The macro instructions used to code an NCP are described in *NCP Generation*, which can be used with the *VTAM System Programmer's Guide* to write a set of statements that both generate an NCP and define that NCP's configuration and functions to VTAM.

VTAM considerations when writing the set of NCP macro instructions are described in this chapter in "Defining NCP Major Nodes."

## Defining the Network to VTAM

Defining the network to VTAM consists of describing the network configuration and its characteristics to VTAM. These descriptions are coded in VTAM definition statements; the coded definitions are then filed in the VTAM definition library.

As part of the network definition, the five types of major nodes in the telecommunication system are defined to VTAM:

- NCPs for local or remote communications controllers, including their attached logical units, physical units, and BSC, and start-stop terminals
- Sets of logical units and physical units on switched lines
- Sets of logical units and physical units attached locally
- Sets of 3270 terminals attached locally
- Sets of application programs that use VTAM

The definition of each node is filed separately (as a member in OS/VS or as a book in DOS/VS) in the VTAM definition library. When VTAM activates a major node, it uses the filed definition of that node as a description of the node's configuration. (In OS/VS, if the major node has been activated previously, a description of the node's configuration is taken from a table that was built when the major node was activated). A major node can be defined anytime following system generation but prior to activating VTAM. Before defining the major nodes to VTAM, it is necessary to understand VTAM's major node and minor node structure.

### *The Major and Minor Node Structure*

Major and minor nodes<sup>1</sup> are the controllable elements of the VTAM network. VTAM definition statements are used to identify all major and minor nodes and to place each node within a hierarchical structure of controllable elements. All major and minor nodes are addressed symbolically using the names assigned when the network is defined to VTAM.

---

<sup>1</sup>The terms *major node* and *minor node* are unrelated to such terms as *host node* and *communications controller node* used in SNA publications.

A major node is a set of controllable elements (minor nodes) in the VTAM network. Each major node structure has the general form:

```
Major Node
  Minor Node 1
  Minor Node 2
  Minor Node 3
  .
  .
  .
  Minor Node n
```

Thus, each major node structure can be controlled as a whole or portions of it can be controlled through the minor nodes within each major node.

One or more minor nodes can be defined for each major node. The name of a minor node is the name assigned to the VTAM statement defining it. The same minor node can be defined in more than one major node. If more than one of these major nodes is activated, only the definition of the minor node associated with the first major node activated is recognized. The types of minor nodes in a major node depends on the type of major node being defined. Figure 3-2 shows the types of major and minor nodes in a VTAM system. Each type of major node is described later in this chapter.

The hierarchical structure of major and minor nodes allows a user to control a group of nodes as a single unit. By activating a major node, a user can activate the minor nodes subordinate to it. However, when activating a minor node, all higher-level nodes must be active before it is activated. Similarly, when deactivating a major node, all minor nodes subordinate to it are automatically deactivated. The activation of major nodes is independent of each other except that a local NCP major node must be activated before an associated remote NCP major node is activated.

### ***Naming Major and Minor Nodes***

All major and minor nodes must be named. The name of each major node is the name of the member (in OS/VS) or book (in DOS/VS) containing the definition statements for that major node. The name of each minor node is supplied on its definition statement. The following rules apply to assigning names:

Duplicate major node names are not permitted.

Duplicate minor node names are not permitted in the same major node.

Two or more major nodes containing duplicate minor-node names cannot be active simultaneously. If the network operator issues an activation request for a major node that contains a duplicate minor-node name, the second minor-node name is ignored.

In any DOS/VS or OS/VS operating system, the following names are assigned to IBM-supplied facilities and must not be used as the names of major or minor nodes: ISTNULL, ISTOLTEP, NETSOL (except for the program itself), PUNS, VTAM, or VTAMSEG. Additionally, in a DOS/VS system, the name TRACE must not be used; in an OS/VS system, ISTATA00 or VTAMTERM must not be used.

Controlling the assignment of node names provides some control over the security of the telecommunication system. See "Telecommunication Security" in Chapter 7 for more information on using node names to protect the system.

### ***Defining Application Program Major Nodes***

One or more major nodes can be defined for application programs. Each major node represents a set of application programs; each minor node represents an individual application program. Each program is defined with an APPL definition statement. To VTAM, an application program is an open access method control block (ACB). The name specified in the APPLID field of the ACB specifies the address of the name stored in the application program that must match the name of an APPL statement.

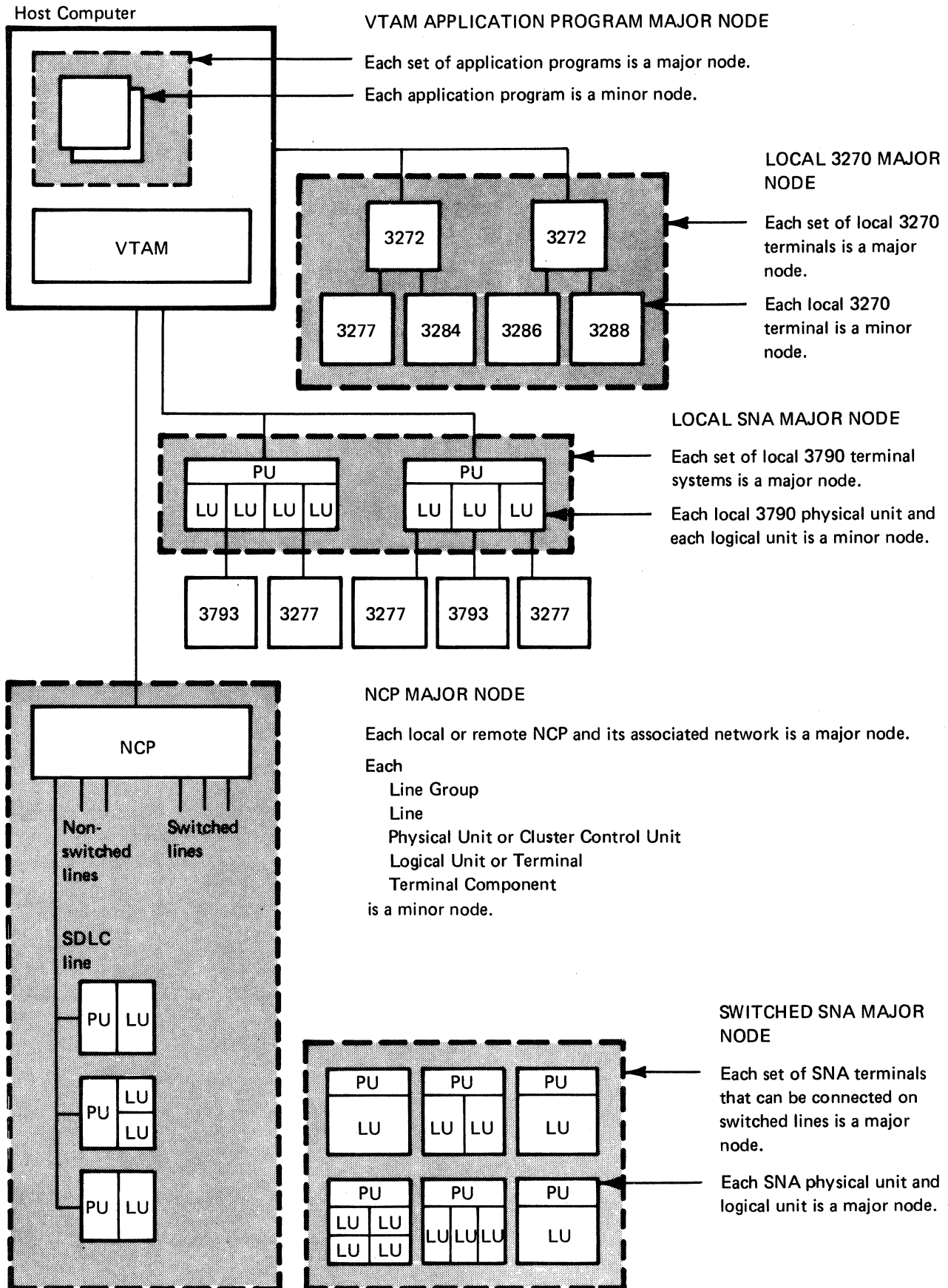


Figure 3-2. Major and Minor Nodes in a VTAM Telecommunications System

APPL statements can be filed separately (as members for OS/VS, or books for DOS/VS) in the VTAM definition library, or they can be grouped in various combinations and filed as sets. The APPL definition statement can provide the following information:

The symbolic name to be located by the APPLID field in the ACB.

The password to be specified by the application program when the ACB is opened.

The specifications of those VTAM facilities that an application program defined by this APPL statement is allowed to use.

The buffer factor for the application program. See “Defining VTAM Buffering” later in this chapter for a description of how buffer limits are established using the buffer factor of the application program.

An application program can be authorized (by its APPL statement) to perform each of the following through VTAM:

Pass terminal connections to another application program.

Initiate connection requests for terminals.

Use the SENDCMD macro instruction to issue VTAM network operator commands (except START and HALT) and the RCVCMD macro instruction to receive VTAM network operator messages. See “Network Operator Control from an Authorized Application Program” in Chapter 4 for a description of this facility.

Alter the VPACING value specified by a logical unit.

Request input data from start-stop or BSC terminals in blocks instead of in messages or transmissions.

An APPL statement must be filed for each unique application program; the same APPL statement can be named by only one open ACB at any one time. An application program can be defined as part of more than one logical set of application programs; that is, the same APPL statement can be filed in more than one member of the VTAM definition library. If more than one set of application programs with a duplicate APPL statement is activated, only one of the APPL statements is recognized.

### ***Defining Local 3270 Major Nodes***

One or more major nodes can be defined for local 3270 terminals. Each major node represents a set of 3270 terminals; each minor node represents an individual 3270 terminal (such as a 3277). Each local 3270 major node is defined with an LBUILD statement; each local 3270 terminal is defined with a LOCAL statement. A local 3270 terminal can be included in more than one local 3270 major node, but if more than one of the major nodes are activated, only the first definition is recognized.

The following information is provided by the LOCAL statements:

The symbolic name of the terminal.

The channel and unit address of the terminal.

The features that are available on the terminal.

The name of the logon description (interpret table) to be used when analyzing logons from the terminal. A terminal’s interpret table is also inspected whenever an INTRPRET macro instruction is issued by an application program for that terminal.

The name of an application program to which VTAM is to automatically transmit a logon whenever the terminal is available for connection. See “Defining VTAM-Initiated Connection” later in this chapter for a description of automatic logon.

Whether the terminal is to be considered active or inactive when its major node is activated.



The buffer limit for the terminal. See “Defining VTAM Buffering” later in this chapter for a description of how buffer limits are established using this specification.

### ***Defining Local SNA Major Nodes***

One or more major nodes can be defined for local 3790s. Each major node represents a set of 3790 Communication Systems. Each minor node represents an individual physical unit or logical unit. The following definition statements define a local SNA major node:

A VBUILD definition statement, which describes the local SNA major node

One or more PU statements, each of which identifies a particular physical unit and describes its characteristics

For each PU statement, one or more LU statements, each of which identifies a logical unit and describes its characteristics

### ***Defining NCP Major Nodes***

One or more NCP major nodes can be defined for each local and remote communications controller used by VTAM. Only one of these major nodes can be active at one time for a given communications controller. The types of minor nodes for an NCP major node are:

**Groups of Lines:** Each group is defined by a GROUP statement.

**Lines:** Each line is defined by a LINE statement.

**SNA Terminals:** The types of minor nodes for SNA terminals are:

*Physical units.* Each physical unit is defined by a PU statement. A physical unit minor node identifies a terminal system controller (such as a 3601), a switched-line access to the communications controller, or, for local NCP major nodes, a remote communications controller. VTAM distinguishes between a remote communications controller attached as an extension of the network and one that is part of a remote station. See Appendix B for a discussion of this distinction. If the PU statement identifies a switched-line access or a remote communications controller, it does not have any logical unit minor nodes.

*Logical units.* Each logical unit is defined by an LU statement. Logical unit minor nodes identify logical units attached on nonswitched lines. Logical units that can be connected on switched lines are defined to VTAM as part of a switched SNA major node.

*Logical unit pools.* Each logical unit pool is defined by an LUPool statement. A logical unit pool minor node is used with switched-line networks and identifies the maximum number of logical units that can be active at one time over the NCP's switched lines.

**BSC and Start-Stop Terminals:** The types of minor nodes for start-stop and BSC terminals are:

*Ports (for switched lines only).* Each port is defined by a TERMINAL statement. A port minor node identifies a start-stop or BSC switched-line dial-in access to the communications controller.

*BSC clusters.* Each cluster is defined by a CLUSTER statement. A cluster minor node identifies a control unit for certain terminals, such as 3270 terminals.

*Terminals.* Each terminal is defined by a TERMINAL statement. A terminal minor node identifies a terminal, such as a 2740 or a 3277, or a remote station, such as a System/3.

*Components.* Each component is defined by a COMP statement. A component minor node identifies a terminal component, such as a printer or a card reader.

The same deck of macro instructions used to generate an NCP can be used as the major node definition by VTAM. Using this deck both for NCP generation and for VTAM

network definition requires additions to the NCP generation macro instructions. These additions include statements<sup>2</sup> and parameters that are used only by VTAM. See Chapter 7 for VTAM requirements for an NCP.

An NCP configuration is defined by filing the NCP generation deck separately as a member in OS/VS or a book in DOS/VS in the VTAM definition library. The member name is thereafter used when addressing the NCP through VTAM.

If an NCP is modified (regenerated), it must be redefined to VTAM. It is redefined by refiling the altered or new NCP generation deck. In OS/VS, it is also necessary to delete the NCP's corresponding member of SYS1.VTAMOBJ. Thus, remote network configurations can be modified without requiring a new or partial generation of the operating system.

Figure 3-3 shows the steps for generating NCP support in a VTAM telecommunication system. The steps are as follows:

1. *Planning the NCP.* Keep VTAM requirements, restrictions, and considerations in mind.
2. *Coding the NCP generation statements.* Include the parameters and definition statements required by VTAM as well as those used to generate the NCP.
3. *Generating the NCP.* Use the statements coded in step 2. Include those parameters and definition statements that are used only by VTAM.
4. *Verifying that the generation is successful.* If the NCP is not generated successfully, correct the generation deck and repeat step 3.
5. *Filing the generation deck.* File the deck (as a member in OS/VS or a book in DOS/VS) in the VTAM definition library. This deck was coded in step 2 and used in step 3. When VTAM activates this NCP, VTAM extracts the information it needs from the filed definition and from the generated NCP itself.

Using the same deck to generate an NCP and to define it to VTAM ensures that the generated NCP agrees with its definition to VTAM. The deck is filed in the definition library using a utility of the operating system.

Generating the NCP prior to filing the deck ensures that the NCP generated is the one that is defined. If the deck is filed first and an error is encountered in the generation process, a corrected deck has to be filed.

### ***Defining Switched SNA Major Nodes***

One or more major nodes can be defined for SNA terminals on switched lines. Each major node represents a set of physical units and their associated logical units. Each minor node represents an individual physical unit or logical unit. The following definition statements define a switched SNA major node:

A VBUILD definition statement, which indicates the start of the defining of a switched SNA major node definition and describes its characteristics to VTAM.

One or more PU statements, each of which identifies a particular physical unit and its characteristics

---

<sup>2</sup>The statements used to generate the NCP can be referred to as *macro instructions* because they are assembled into communications controller instructions. VTAM, however, uses these same statements unassembled. In this publication, therefore, they are referred to as *statements*, not macro instructions.

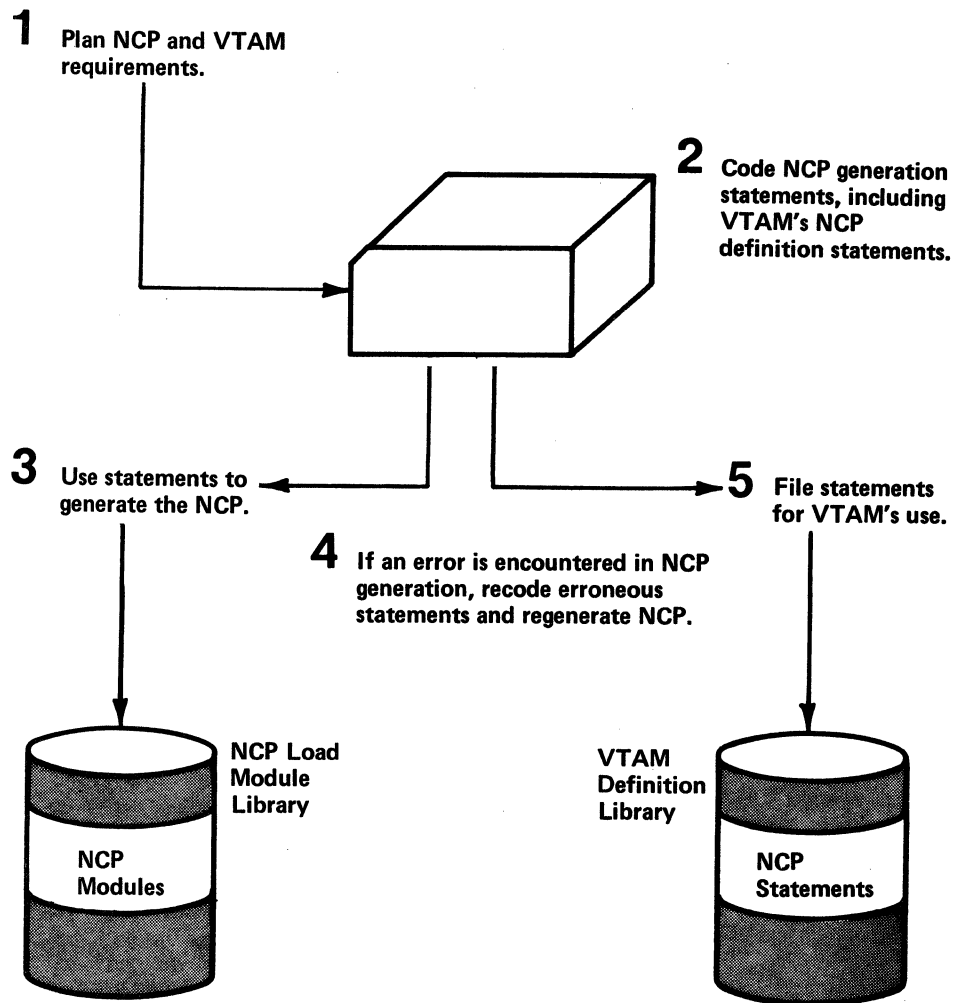


Figure 3-3. Generating NCP Support in a VTAM Telecommunication System

For each PU statement, one or more PATH statements, each of which describes how a dial-out operation will be carried out (the telephone number to call and the number of redial attempts to be made)

For each PU statement, one or more LU statements, each of which describes the characteristics of a logical unit associated with the physical unit

## Defining Connection Procedures

When a terminal is activated, either when VTAM is started or when the network operator enters a command requesting activation, VTAM performs any preparation required for connection, such as transmitting an Activate Physical Unit command, with no network operator or VTAM application program action. For terminals on switched lines, the active status is logical rather than physical; physical connection to VTAM is not actually established until a dial-in or dial-out operation takes place. Once a terminal is active, a request for connection to a VTAM application program can be made by:

The terminal (a logon)

VTAM (an automatic logon)

A VTAM application program (a simulated logon, passed connection, or acquisition)  
The network operator (VARY LOGON command)

Except for a VTAM application program's request to acquire a terminal, a request for connection between a VTAM application program and a terminal takes place in two stages:

1. VTAM receives the request and notifies the application program.
2. The VTAM application program either accepts or rejects the terminal by making the appropriate request of VTAM.

VTAM passes on a request for connection to an application program by either scheduling the program's LOGON exit routine, if one has been designated, or completing an OPNDST macro instruction that specifies OPTCD=ACCEPT. The application program issues an OPNDST macro instruction that specifies OPTCD=ACCEPT to accept the connection request or a CLSDST macro instruction to reject the request. Chapter 5, "Writing a VTAM Application Program," discusses how a VTAM application program connects to terminals.

### ***Defining Terminal-Initiated Connection***

A user can allow a terminal to request connection to a VTAM application program. Defining this kind of connection requires defining logon formats to VTAM, the VTAM application program, and the terminal. This section describes defining these procedures for logical units; see Chapter 8 for a description of defining these procedures for local 3270, BSC, and start-stop terminals.

### **Types of Logical Unit Logons**

VTAM receives a logon from a logical unit as either a field-formatted Initiate Self command or a character-coded logon. The logon contains the name of the application program with which connection is requested, an indication of the type of application program activity required (for example, batch or interactive), and data that can be passed to the VTAM application program as the logon message. With some SNA terminals, a user may not need to know the exact format of a logon because VTAM and the terminal or the IBM-supplied terminal system program formats the message appropriately.

In general, logical units associated with programmable controllers can send field-formatted logons; some can also send character-coded logons. In general, logical units associated with non-programmable SNA terminals send character-coded logons. See the programming publications for the particular SNA terminal product to determine whether to plan for field-formatted or character-coded requests or both. Figure 3-4 provides an example of the types of requests for several IBM SNA terminal products.

For an installation that uses only field-formatted logons, the user may not need to understand character-coded logons. For an installation that uses only character-coded logons, the user must understand field-formatted logons only to the extent that a character-coded logon is converted by VTAM into a field-formatted logon.

A field-formatted logon is processed by the formatted system services (FSS) of VTAM. A character-coded logon is processed by the unformatted system services (USS) of VTAM. *System services* are performed by the *system services control point (SSCP)*, which is the part of VTAM that provides such services as processing requests for connection and disconnection. VTAM converts a character-coded request into a field-formatted request before it is processed.

Type of SNA Terminal	Form of Logon/Logoff That Can Be Used	In the LU Statement, Specify SSCPFM=
3270	Character-coded (required)	USS3270
3600	Field-formatted	FSS
3767	Character-coded (required)	USSSCS
3790	Field-formatted	FSS

Figure 3-4. The Form of Logon/Logoff Required or More Commonly Used by Several SNA Terminals

### Defining Field-Formatted Logons

A field-formatted logon is formatted when VTAM receives it. VTAM determines that it is a connection request for a particular VTAM application program and passes the request to the VTAM application program by completing an outstanding OPNDST (with OPTCD=ACCEPT) or by scheduling the program's LOGON exit routine. No special action is required when the network is defined to VTAM to prepare for field-formatted logons. The logical unit specifies the name of the VTAM application program that is requested (the name of an APPL statement) in the field-formatted logons.

In general, field-formatted logons are made by programmable SNA terminals. Depending on the product, the program may be supplied by IBM or the user. The program formats the request into fixed fields, based either on a logon from an operator or on some predefined program logic. In either case, the programmer of the terminal program must know the name of the VTAM application program with which connection is requested. To specify that logons from a logical unit will be field formatted, each LU statement should specify SSCPFM=FSS or omit the SSCPFM operand when defining the network to VTAM. Figure 3-5 shows a field-formatted logon.

### Defining Character-Coded Logons

Character-coded logons are used by SNA terminals that are not programmable. The terminal operator directly keys in a request for connection to a specific VTAM application program, indicating an application mode (such as interactive or batch) and specifying data (a logon message) to be passed to the VTAM application program. VTAM receives the character-coded logon, converts it to a field-formatted logon, and processes it as a field-formatted logon. A user has a great deal of flexibility in defining the format and content of what the terminal operator can enter as a logon, either using an IBM-supplied definition table or supplying its own. To specify that logons will be character-coded, each LU statement should specify SSCPFM=USSSCS or, for an SNA 3270 terminal, SSCPFM=USS3270, when defining the network to VTAM. Figure 3-6 shows a character-coded logon.

### Using the IBM-Supplied Character-Coded (USS) Definition Table

For logical units that send character-coded logons, IBM defines logon formats and supplies a related definition table that VTAM uses to convert them to field-formatted logons. To use this IBM-supplied definition table, the user specifies SSCPFM=USSSCS or SSCPFM=USS3270, as appropriate, in the LU statement during VTAM definition.

The IBM-supplied USS definition table (ISTINCDT for DOS/VS and OS/VS2 MVS, or ISTINADT for OS/VS1 and OS/VS2 SVS) requires that the terminal operator enter a logon in this form:

```
LOGON APPLID(application program name) LOGMODE(logon mode) DATA(logon message)
```

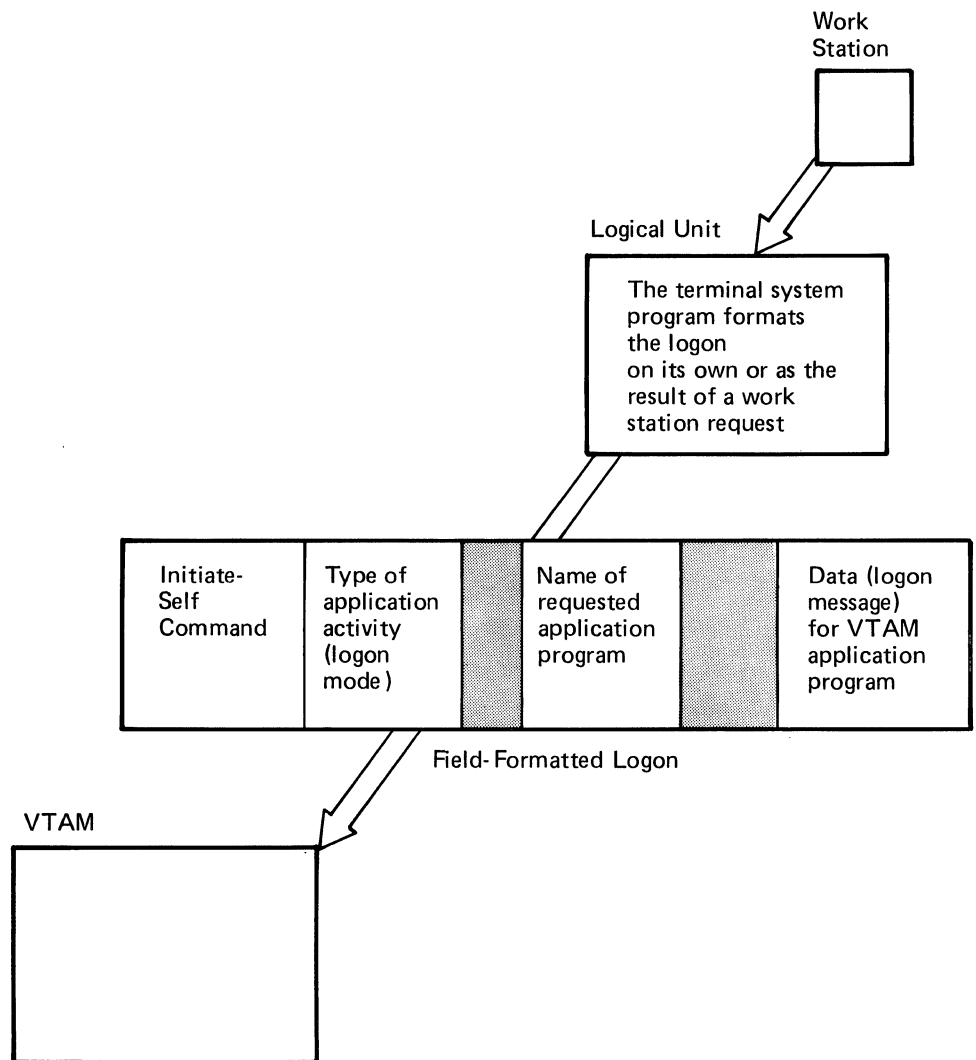


Figure 3-5. A Field-Formatted Logon

The logon mode is the type of application work planned, such as batch or interactive (defining a logon mode table is described later in this chapter). If the IBM-supplied logon is entered in lowercase, VTAM translates it to uppercase. If a character-coded logon is incorrectly entered, VTAM sends an appropriate message to the terminal operator.

**Defining a Character-Coded (USS) Definition Table**

A user can define a logon format and associated definition table instead of using the logon format and definition table supplied by IBM. This allows a user to select:

Its own character translation table instead of the table supplied by IBM. The user's table can translate numeric characters to alphabetic characters, for example.

A BAL format rather than a PL1 format. For example, the terminal operator can enter:

```
LOGON APPLID=application name,LOGMODE=logon mode,DATA=logon message
```

Replacements for the IBM-supplied parameter names.

Default values for the application program name, logon mode, and logon message.

Alternate text for USS error messages.

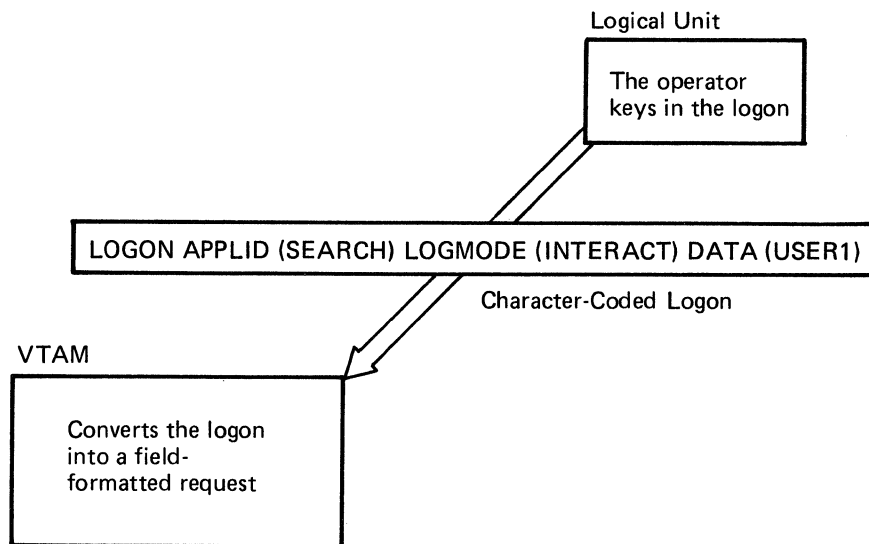


Figure 3-6. A Character-Coded Logon

As shown in Figure 3-7, to define a logon format and definition table, the user specifies the name of the definition table in the USSTAB operand of the LU statement when the network is defined to VTAM. The USS definition table is created by following this procedure:

1. Code a USSTAB macro instruction to indicate the beginning of the USS definition table. Use the TABLE operand to specify a character translation table if one is to be substituted for the IBM-supplied character translation table.
2. For each command (verb) that a terminal operator can enter, code a USSCMD macro instruction that specifies the verb, the replacement USS verb (LOGON or LOGOFF), and the format (PL1 or BAL) of the user-entered logon.
3. For each USSCMD macro instruction, code a USSPARM macro instruction for each positional or keyword parameter on the user-entered logon. USSPARM specifies the parameter, the USS keyword to identify the parameter value, and a default value to be used if the parameter is not entered.
4. For each USS message to be replaced, code a USSMSG macro instruction that specifies the number of the message and the replacement text.
5. Code a USSSEND macro instruction to indicate the end of the definition table.
6. Assemble, link-edit, and load the resulting module into the VTAM load module library.

Figure 3-8 shows an example of how a user-defined USS definition table can translate a terminal operator's logon.

### Defining and Using an Interpret Table

For either field-formatted or character-coded logons, a user can define and use an interpret table. An interpret table provides an additional translation of the application program name to which connection is requested. The name of the interpret table is coded on the LOGTAB operand of the LU statement when the network is defined to VTAM. The interpret table is designed primarily for use in connection procedures for local 3270, BSC, and start-stop terminals. It is described in Chapter 8 under "Specifying Interpret Tables."

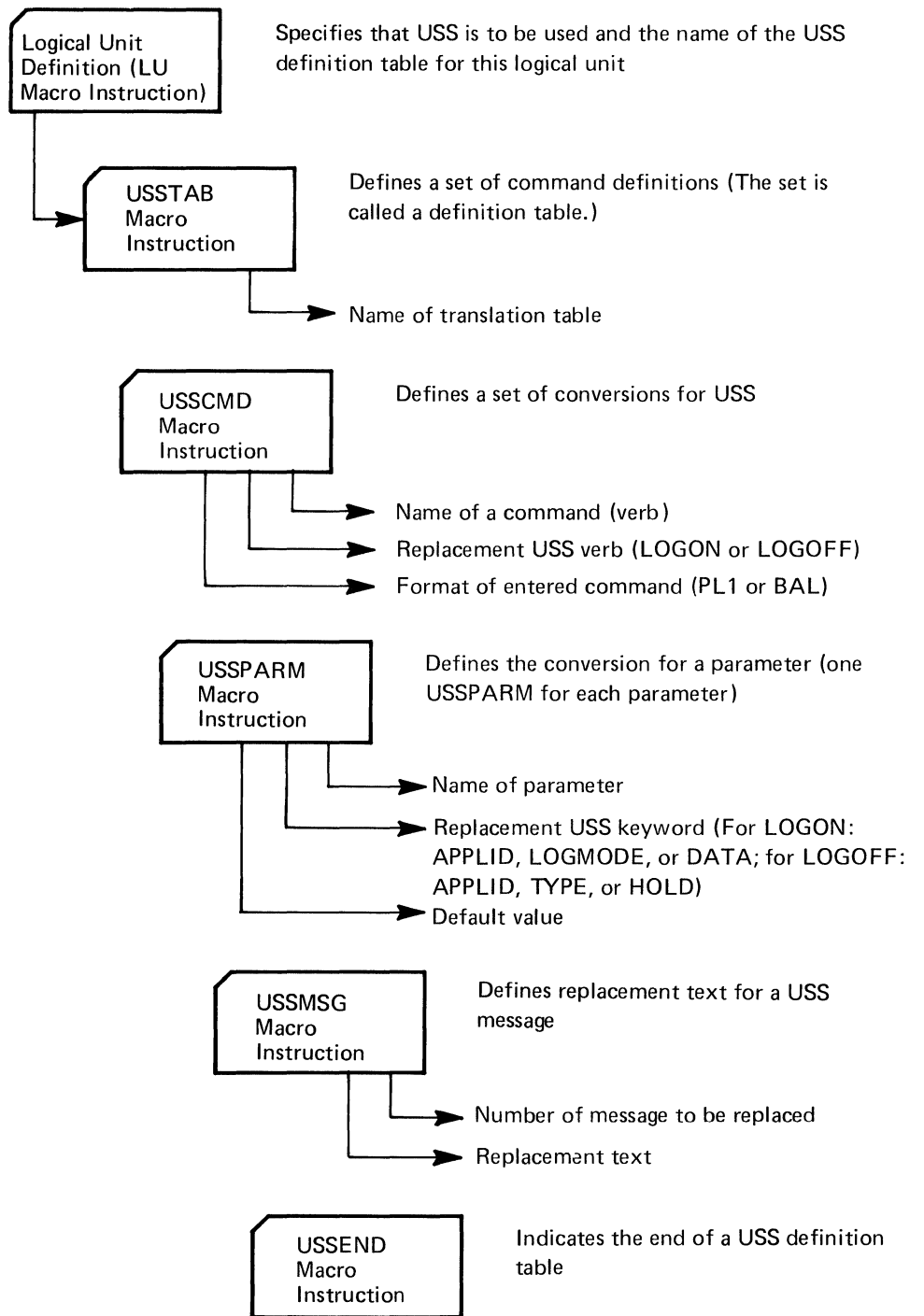


Figure 3-7. Defining a USS Definition Table



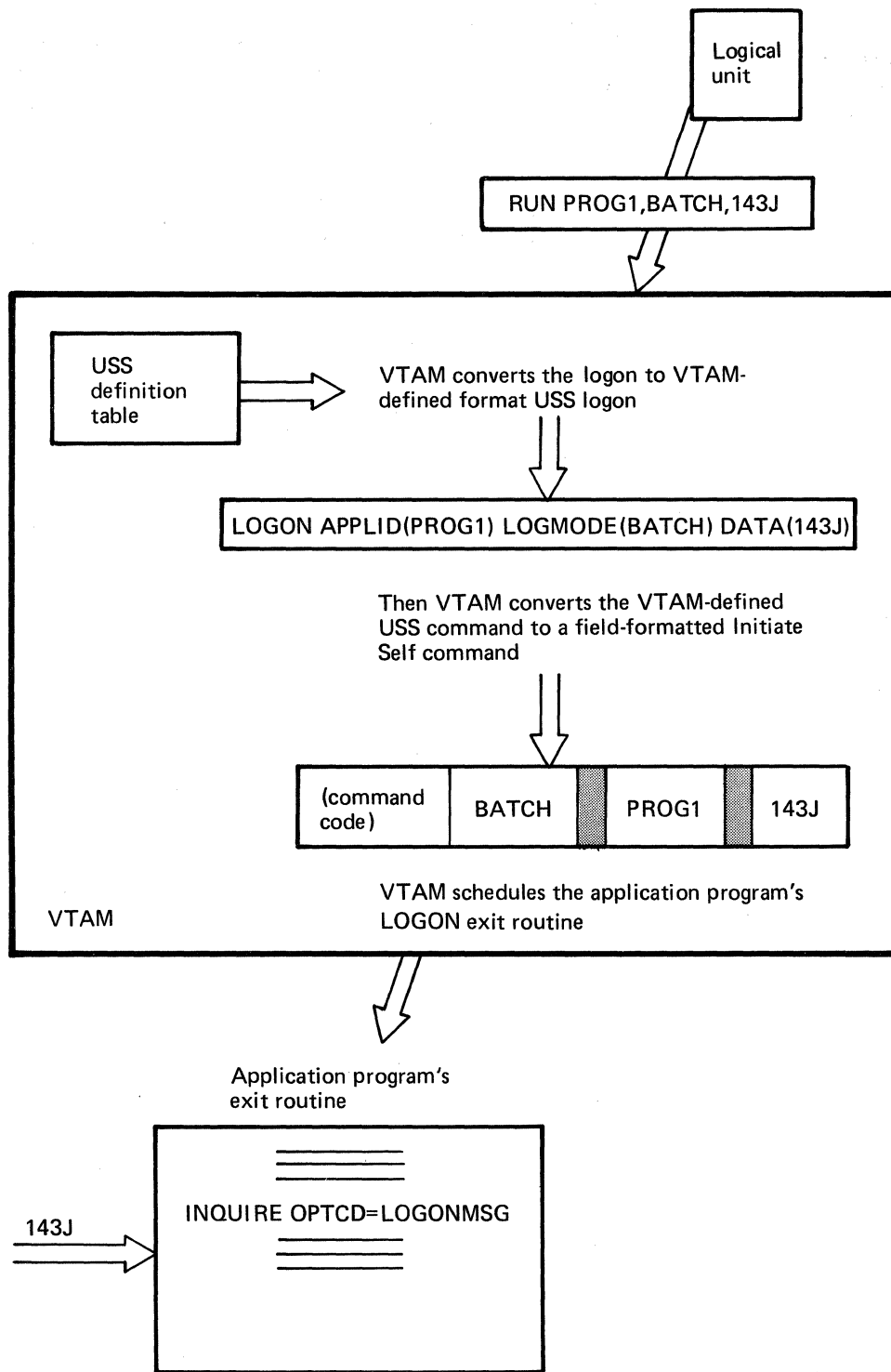


Figure 3-8. Example of VTAM Handling a Character-Coded (USS) Logon

***Defining VTAM-  
Initiated Connection  
(Automatic Logon)***

A user can specify that a logon is to be automatically generated on behalf of a terminal (a logical unit or a local 3270, BSC, or start-stop terminal) for a selected application program whenever that terminal is active and not already connected or queued for connection to another program. This specification is made by coding the name of the program in the LOGAPPL operand of the terminal's definition statement. The name is the same as the name specified for the application program in its APPL definition statement. (Local 3270, BSC, and start-stop terminals can be automatically logged on to the network solicitor, that handles logons from these terminals. NETSOL is the name of the IBM-supplied network solicitor and can be specified in the terminal's LOGAPPL operand. See Chapter 8 for details on the network solicitor.) When defining a VTAM-initiated connection, it should be noted that the VTAM application program cannot receive a logon message with the logon.

For logical units that may call in on switched lines, VTAM holds an automatic logon designation for an activated logical unit pending until a dial-in operation for the logical unit occurs. VTAM then schedules the designated LOGON exit routine or completes an OPNDST that specifies OPTCD=ACCEPT.

The network operator can change the automatic logon specification. See "Defining Network Operator-Initiated Connection," below for a description of this facility.

***Defining VTAM  
Application Program-  
Initiated Connection***

A VTAM application program can initiate a request for connection of a terminal to itself or to another application program. To request that a terminal be connected to itself, a VTAM application program can:

Directly acquire the terminal by issuing an OPNDST macro instruction that specifies OPTCD=ACQUIRE

Cause its LOGON exit routine to receive a request for connection from another part of the program by issuing a SIMLOGON macro instruction

A VTAM application program can request that a terminal currently connected to it be passed to another active VTAM application program. This is done by issuing a CLSDST macro instruction that specifies OPTCD=PASS with the name of the program to which the terminal is to be passed specified in a designated area of the program issuing the CLSDST. Optionally, a logon message can also be passed. As a result of this CLSDST, VTAM completes an outstanding OPNDST for that terminal or, if there is no outstanding OPNDST, schedules the receiving program's LOGON exit routine. A user must authorize a program to use this procedure by specifying AUTH=PASS on the program's APPL statement during VTAM definition.

Considerations for using these forms of application program-initiated connection are discussed below.

***Acquiring Connection  
(OPNDST with OPTCD=  
ACQUIRE)***

This form of connection might be useful when:

The VTAM application program determines whether or not a shareable resource, such as a master display terminal or printer, is in use by another program, and acquires it if it is not.

An application program requires an output-only terminal, which cannot initiate a connection request.

As an alternative to a VTAM-initiated (automatic) logon, a terminal is to be routinely assigned to the program for the duration of its being online.

It is not necessary for all logons to be processed by the program's LOGON exit routine (if it is, a SIMLOGON should be used).

The user must authorize acquisition of terminals by specifying AUTH=ACQ on the program's APPL statement when the network is defined to VTAM.

### **Simulating a Logon (SIMLOGON)**

This form of connection might be useful when:

The user wants to have all connections occur at the same place in the program (in the LOGON exit routine). This might be because the LOGON exit routine keeps track of how many or which terminals are connected to the program.

The user wants all terminals to be connected in the same manner to an application program, using a correct logon message. (Logons automatically initiated by VTAM cannot specify a logon message.)

It is uncertain whether or not a terminal will be available for connection. The SIMLOGON macro instruction can request that it be queued if the terminal is unavailable (active but already connected to another program). When the terminal becomes available, the program's LOGON exit routine is scheduled.

The user must authorize simulated logons by specifying AUTH=ACQ on the program's APPL statement when the network is defined to VTAM.

### **Passing Connection to Another Program (CLSDST with OPTCD=PASS)**

This form of connection might be useful when a set of VTAM application programs are to be used in sequence. For example, an installation maintenance program may produce data that is used by a related program and then printed on a terminal.

The user must authorize passing connection to another program by specifying AUTH=PASS on the program's APPL statement when the network is defined to VTAM.

### **Defining Network Operator-Initiated Connection**

The user can plan for a network operator to initiate connection for a terminal (a logical unit or local 3270, BSC, or start-stop terminal) to an application program. This could be done routinely or as the result of special occurrences, such as an authorized user's calling and requesting connection of terminals to a particular VTAM application program. The network operator can initiate a connection request with the VARY command, specifying the names of both the terminal and the application program and, for SNA terminals, the logon mode. If the network operator initiates the connection request, a logon message cannot be specified.

The VARY command modifies any automatic logon designation for the terminal. Once the command is issued, the terminal specified in that command is automatically logged on to the application program whenever it is available unless that program has specifically released it. The newly designated automatic-logon specification remains in effect until one of the events described below occurs:

**In OS/VS and DOS/VS Release 33**, the newly designated automatic-logon specification remains in effect until (1) the network operator issues another VARY command for that terminal and specifies another application program or (2) the major node containing that terminal is deactivated and reactivated with a cold restart. When the major node is reactivated cold, the automatic logon conditions specified when the VTAM network was defined are in effect.

**In DOS/VS Release 32**, the newly designated automatic-logon specification remains in effect until (1) the network operator issues another VARY command for that terminal and specifies another application program or (2) the major node containing that terminal is deactivated. When the major node is reactivated, the automatic logon conditions specified when the VTAM network was defined are in effect.

***Defining Logon Modes***

Part of a logon for a logical unit (whether initiated by VTAM, an application program, the network operator, or the logical unit itself) is a symbolic name, called a *logon mode name*, that indicates a set of communication rules that the VTAM application program and the logical unit agree to follow. (VTAM does not necessarily enforce these rules; the VTAM application program, however, may not work properly if it fails to conform to the rules agreed upon.) These rules generally correspond to modes of application program operation, such as interactive or batch operation. For some SNA terminal products, the



logon mode is preestablished as, for example, always interactive or always batch. For others, the logon mode can be varied, and the connection initiator must select the appropriate logon mode.

Each logon mode name indicates an entry in a *logon mode table* associated with the logical unit that defines a set of rules (called *session parameters*). When an application program accepts a logon, it can request the session parameters associated with the pending logon, the session parameters associated with a logon mode name it specifies, or it can define its own set of session parameters and request them. When an application program acquires a logical unit (OPNDST with ACQUIRE), it can request the session parameters associated with a logon mode name it supplies or it can define its own set of session parameters and request them.

A user can create logon mode tables in addition to or instead of the IBM-supplied logon mode table. Each table contains one or more entries; each entry describes a logon mode name. To create a logon mode table, the user uses **MODETAB**, **MODEENT**, and **MODEEND** macro instructions as shown in Figure 3-9. The **MODETAB** and **MODEEND** macro instructions specify a group of logon mode names and each **MODEENT** defines one logon mode name. Each logon mode table is assembled and filed separately (as a member in OS/VS or book in DOS/VS) in the VTAM load module library. The name assigned to the load module is specified in the **MODETAB** operand in the LU statement for each logical unit with which this table is to be associated. Each entry in that table has both a logon mode name, such as **INTERACT**, and a related set of session parameters (the *logon mode*) associated with that name.

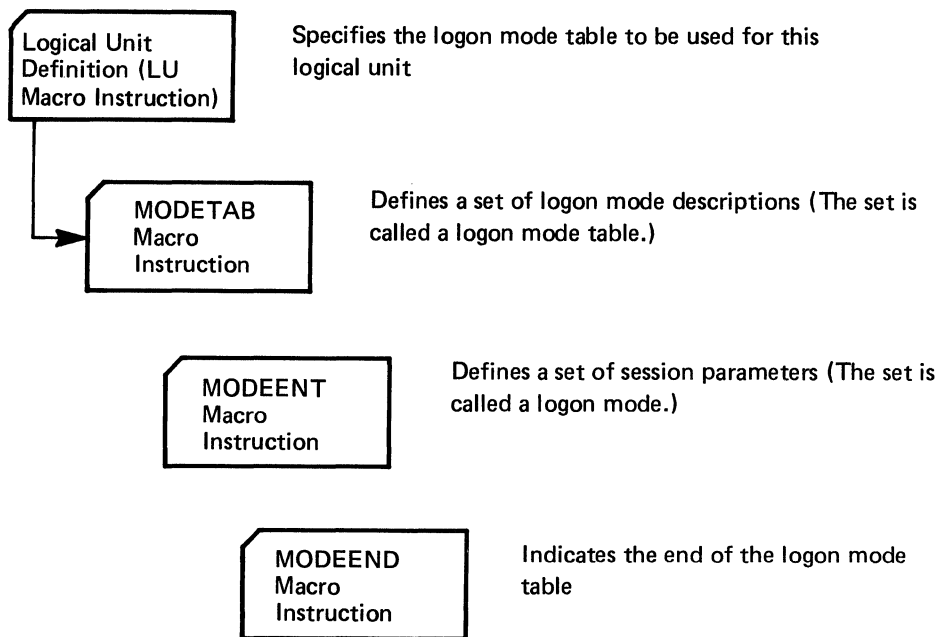


Figure 3-9. Defining a Logon Mode Table

## Defining Disconnection Procedures

A user must also plan the procedure by which a terminal is disconnected. A request to disconnect a terminal can come from:

- The terminal (in a logoff)
- The VTAM application program
- The network operator
- VTAM (because of a hardware or software error)

If a terminal is still active after it has been disconnected, it may be available for connection to another VTAM application program.

### *Disconnection Requested by the Terminal*

For local 3270, BSC, and start-stop terminals, the VTAM application program may search each data message from the terminal for a predesignated character or character-string indicating a request for disconnection. VTAM does not provide any special facility for logoff from local 3270, BSC, and start-stop terminals. When the application program receives a logoff from a local 3270, BSC, or start-stop terminal, it may issue a CLSDST macro instruction, causing VTAM to disconnect the terminal from the program.

For SNA terminals, VTAM provides a logoff facility similar to the logon facility described previously in this chapter in "Defining Connection Procedures." A logical unit can send either a field-formatted or character-coded logoff. The field-formatted logoff is a Terminate Self command; the character-coded logoff is a character string that VTAM converts into a Terminate Self command. Consult the programming publications for each particular terminal product to determine the kind of logoff used (field-formatted or character-coded or either).

If a character-coded logoff is used, the user can either use the IBM-defined logoff format, or it can define its own logoff format by defining a USS definition table, as previously described in "Defining a Character-Coded (USS) Definition Table." The IBM-defined logoff format is:

```
LOGOFF APPLID(application name) TYPE(COND or UNCOND) HOLD(YES or NO)
```

The logoff parameters, whether they arrive in a Terminate Self command or in a character-coded logoff defined by IBM or the user, specify:

- The VTAM application program from which the logical unit is to be disconnected
- Whether disconnection is mandatory (UNCOND) or at the discretion of the application program (COND)
- Whether the physical unit with which this logical unit is associated is also to be disconnected if this is the last remaining logical unit that is connected

### **Conditional or Unconditional Disconnection**

The VTAM application program from which disconnection is requested gets control as a result of the logoff. Its action depends on the conditional or unconditional parameter.

If the logoff specifies conditional termination, VTAM schedules the application program's LOSTERM exit routine. The program can then ignore the request or disconnect the logical unit by issuing a CLSDST macro instruction. This kind of logoff allows the application program to exchange some final messages with the logical unit. If the logoff specifies unconditional termination or omits the parameter, VTAM stops all communication with the logical unit and then schedules the program's LOSTERM exit routine with an indication that the logical unit must be disconnected.

## **Holding the Physical Unit Connection**

Although not apparent to the VTAM application program, VTAM establishes a connection between itself and each physical unit and logical unit before it establishes a connection between an application program and the logical unit associated with the physical unit. For physical units on nonswitched lines, this connection is established between VTAM and a physical unit when the physical unit is activated. For physical units on switched lines and those that are or locally attached, the connection between VTAM and the physical unit is not established until a VTAM application program or a logical unit associated with a physical unit requests a connection between a VTAM application program and the logical unit. Once the physical unit on a switched line is connected to VTAM, VTAM maintains the connection as long as one of its logical units is connected, unless the user specifies otherwise. The user has the following choice:

- When defining the physical unit to VTAM the user specifies whether the physical unit is to be disconnected as soon as the last of its logical units has become disconnected (DISCNT=YES) or whether this is to be decided by the logical units (DISCNT=NO). (DISCNT=YES is the default.)
- If DISCNT=NO is specified during VTAM definition, each logoff request can specify whether or not the physical unit connection is to be held if this is the last logical unit that is connected. This is done by specifying the Last or Not Last indicator in a field-formatted Terminate Self command or HOLD(YES or NO) in a character-coded logoff. The default for a field-formatted command is to hold the connection; the default for a character-coded logoff is not to hold the connection if this is the last remaining logical unit.
- If the logoff for a logical unit specifies that the physical unit connection is to be held, the physical unit can be disconnected by a Ready-to-Go-on-Hook (Request Disconnect) command from the physical unit (which might result from timing logic in a program associated with the terminal, or from the action of an operator at the terminal or terminal controller represented by the physical unit).

The user might want to hold the connection to a physical unit when logical unit connections to application programs are expected to be frequent but of short duration to allow periods of time during which no logical unit is in session with the host. For physical units on switched lines, holding the physical unit connection allows control over the frequency of dial-in or dial-out operations for the physical unit.

Disconnection of the physical unit from VTAM is different for physical units on nonswitched lines and physical units on switched lines or that are locally attached. When a physical unit on a nonswitched line is disconnected from VTAM, it is also deactivated. When a physical unit on a switched line or one that is locally attached is disconnected, it remains active.

## **Using the Request Shutdown Protocol**

A logical unit can also use a Request Shutdown command to request disconnection. This command advises the VTAM application program that the logical unit is ready to complete its work. The application program can send any final messages to the logical unit and then issue a Shutdown command. When the application program in turn receives a Shutdown Complete reply, it can disconnect the logical unit using the CLSDST macro instruction.

## ***Disconnection Requested by the VTAM Application Program***

A VTAM application program can decide that communication with a connected terminal has ended and that the terminal is to be disconnected. This might be the normal procedure for disconnection on termination of batch output to a terminal. The VTAM application program issues a CLSDST macro instruction, causing VTAM to logically disconnect the program from the terminal. OPTCD=RELEASE can be specified to allow the terminal to be available for connection to any other program. OPTCD=PASS can be



specified to notify a specified application program that the terminal is available for connection. The user must authorize a VTAM application program to use OPTCD=PASS when the network is defined to VTAM.

A VTAM application program can also issue a CLSDST macro instruction after receiving a Shutdown Complete command from a logical unit as discussed previously in "Using the Request Shutdown Protocol."

***Disconnection  
Requested by the  
Network Operator***

The network operator can deactivate a terminal (or a higher level of major or minor node) normally or immediately. Normal deactivation does not result in a disconnection request on behalf of a terminal; it simply indicates to VTAM that the next time the terminal becomes disconnected from a VTAM application program, it is to become inactive. Immediate deactivation, however, acts as a logoff on behalf of a terminal. The VTAM application program's LOSTERM exit routine is scheduled, and the program should then issue a CLSDST macro instruction to disconnect the terminal, allowing the terminal to become inactive. This procedure might occur most frequently when the network operator has become aware of a problem with the terminal or network of which the VTAM application program is not aware.

***Disconnection  
Requested by VTAM***

VTAM requests that a terminal be disconnected when certain hardware and software errors occur that cause loss of contact with that terminal. VTAM schedules the application program's LOSTERM exit routine indicating the reason for the disconnection request (such as failure in the network path between a communications controller and a remote terminal). The application should issue a CLSDST macro instruction to disconnect the terminal.

***Effect of  
Disconnection on  
Automatic Logon***

Whatever procedure an application program follows for logoffs, when a terminal with an automatic logon specification is disconnected, VTAM usually attempts to queue the terminal to the application program named in that specification. Some conditions can arise in which a terminal is not immediately queued in accordance with the automatic logon specifications. For a description of these conditions, see the discussion on disconnection in Chapter 5, "VTAM Application Programs."

## **Authorization, Accounting, and Logon-Interpret Exit Routines**

VTAM provides for two types of user-written exit routines. One type consisting of RPL and EXLST exit routines is coded and identified in each application program that uses VTAM. These exit routines are executed as part of the application program and are under the control of the application program. A second type consisting of authorization, accounting, and logon-interpret exit routines is coded and included in VTAM as part of VTAM definition. These routines are executed as part of VTAM and are not under the control of application programs. Information on the RPL and EXLST exit routines is provided in Chapter 5. The remainder of this section provides information on coding and using authorization, accounting, and logon-interpret exit routines, and on including them in VTAM.

VTAM provides for one authorization and one accounting exit routine. If the user does not provide its own exit routines, IBM-supplied modules are used. One logon-interpret routine can be coded for each LOGCHAR macro instruction used to define an interpret table (described in Chapter 8), although the same routine can also be used for more than one LOGCHAR instruction. Logon-interpret routines are not provided with VTAM; if these routines are needed, they must be supplied by the user. Each type of exit routine is described below.

## ***Authorization Exit Routine***

VTAM provides an exit that permits the user to authorize connections between application programs and terminals. (Authorization in the exit routine is in addition to that performed by VTAM.) This exit routine is scheduled whenever a terminal is to be queued for logon to, connected to, or disconnected from an application program. Thus, the routine at this exit is executed whenever a connection is to be made or broken as a result of an OPNDST, SIMLOGON, or CLSDST macro instruction in the application program, or whenever a terminal is to be queued as the result of a logon.

The following information is available when this exit routine is entered:

The type of connection or disconnection request that has been made.

The names of the terminal and the application program to be connected, disconnected, or queued. Also, if the operation is a logon resulting from a CLSDST macro instruction with the PASS option, VTAM also provides the name of the application program issuing the pass request.

Using this input, the authorization routine can determine whether each connection, disconnection, or logon should be processed by VTAM. For example, each request might be compared against a predefined, user-specified table of valid or invalid requests. The results of this examination are then returned to VTAM. If the request is determined to be valid, it is completed by VTAM. If it is invalid, it is rejected by VTAM. Output from this routine must include whether the request is valid or invalid.

In DOS/VS, the authorization routine is included in VTAM by cataloging it into the core image library. In OS/VS1 and OS/VS2 SVS, it is included by link-editing it as a single load module into the VTAM load module library. In OS/VS2 MVS, it is included by link-editing it as a single load module into the SYS1.LPALIB data set. This routine replaces an IBM-supplied module that treats all requests as valid.

In planning the authorization routine, a number of factors must be considered:

- The exit routine is executed in the supervisor state under VTAM's protection key. Therefore, errors within the routine may cause damage to VTAM's control blocks and modules. Also, security violations can occur because such a routine has access to much of the VTAM partition and private address space.
- The exit routine is executed under the task for which the request is being authorized (for example, the application program that issues the connection request). If the exit routine is abnormally terminated, this task may be terminated also.
- The exit routine is executed inline with VTAM processing. Therefore, performance may be degraded if the routine requires lengthy processing time. While this routine is being executed, no new connection, disconnection, or logon requests are processed by VTAM, and requests involving VTAM's VARY command are not processed. Therefore, system waits (such as for disk I/O) should be avoided.
- The exit routine is notified of pass requests. (The pass option is an option of the CLSDST macro instruction; this option is used by VTAM's network solicitor, the port solicitor (this facility supports start-stop and BSC switched networks), and the interface to the Telecommunications Access Method (TCAM) and can be used by application programs.) The network solicitor uses the pass option to pass valid terminal-initiated logons to active application programs. If the network solicitor is used to monitor terminals for logons, the authorization routine should be designed to process the validation of pass requests involving the network solicitor.
- The exit routine is notified if VTAM's network solicitor or the Terminal Online Test Executive Program (TOLTEP) attempts to connect a terminal. An authorization routine should be designed to process these requests. (See "Serviceability Aids" in Chapter 6 for a description of TOLTEP.)

## ***Accounting Exit Routine***

VTAM provides an exit that permits a user to maintain accounting information about connections. This exit routine is scheduled whenever a terminal and an application program are connected or disconnected. Thus, the routine is executed each time the application program issues an OPNDST or CLSDST macro instruction.

When the routine is entered, the name of the program, the name of the terminal, and the type of request (connection or disconnection) is available. Using this input, an accounting routine can note and record the time a connection is initiated. Upon reentry (when that connection is being broken), the routine can note the time of the disconnection. The difference between these two times is the connection time for the terminal and that application program. Note that connection time is an approximate value affected by such factors as the running time of the application program, the paging rate, and the operating system setting the application non-dispatchable.

In DOS/VS the accounting routine is included in VTAM by cataloging it into the core image library. In OS/VS1 and OS/VS2 SVS, it is included by link-editing it as a single load module into the VTAM load module library. In OS/VS2 MVS, it is included by link-editing it into the SYS1.LPALIB data set. This routine replaces an IBM-supplied module. If the user does not replace the IBM-supplied accounting exit routine, no accounting statistics are gathered.

In planning an accounting routine, a number of factors must be considered:

- The exit routine is executed in the supervisor state under VTAM's protection key. Therefore, errors within the routine may cause damage to VTAM's control blocks and modules. Also, security violations can occur, because such a routine has access to much of the VTAM partition and private address space.
- The exit routine is executed under the task about which the accounting information is being collected (for example, the application program that issues the connection request). If the exit routine is abnormally terminated, this task may be terminated also.
- The exit routine is executed inline with VTAM processing. Therefore, performance may be degraded if the routine requires lengthy processing time. While this routine is being executed, no new connection, disconnection, or logon requests are processed by VTAM, and requests made through the VARY command are not processed. Therefore, system waits (such as for disk I/O) should be avoided.
- The exit routine is notified of connection and disconnection requests involving terminals and IBM-supplied facilities. These facilities are TOLTEP, the network solicitor, the port solicitor (this facility supports start-stop and BSC switched networks), and the interface to the Telecommunications Access Method (TCAM). The user-coded accounting routine should be designed to process requests involving these facilities.

## ***Logon-Interpret Routines***

Details on the logon-interpret routines are provided under "Specifying Interpret Tables" in Chapter 8. They are mentioned here to indicate that they can be thought of as installation-level routines as opposed to application-program-level routines. The logon-interpret routines would probably be planned and coded as part of VTAM definition. Also, because these routines can be used to prohibit logons, they can be used in conjunction with the authorization exit routine to control connections.

## Defining VTAM Start Options

When VTAM is started, options can be used to define the initial VTAM network and to select optional VTAM facilities. These start options can be specified by the operator as part of the START command (in OS/VS only) or as a response to the prompting of VTAM (in DOS/VS or OS/VS). The options can also be predefined and filed in the VTAM definition library. (See "Starting VTAM" in Chapter 4 for information on specifying options from the system operator's console.)

The VTAM start options tailor VTAM to the user's needs each time the telecommunication system is started. Predefining start options relieves the network operator of this activity. In addition, more than one version of the start options can be predefined, each version specifying a different VTAM configuration. With different sets of predefined options, the user can initialize a particular VTAM system merely by selecting the appropriate set of options.

The predefined start options are stored under the member or book name ATCSTRxx where xx is a two-character identification created by the user. A data set must be filed under ATCSTR00 even if it does not contain any options. Other ATCSTRxx data sets can be created, but the specific data set required is specified by the user when VTAM is started. The following start information can be supplied in the ATCSTRxx data set:

Which major and minor nodes are to be traced by the VTAM trace facility

Which major nodes are to be activated during start processing (VTAM initialization)

Whether the status (active or inactive) of major nodes is to be recorded (OS/VS and DOS/VS Release 33 only)

Whether minor nodes within an activated major node are to be given their initial status, or their predeactivation status as defined by their configuration restart data sets (OS/VS and DOS/VS Release 33 only)

The size of VTAM storage pools and in OS/VS, which pageable storage pools should be fixed

The maximum number of major nodes active at any one time

Whether certain network operator messages are to be suppressed

Whether prompting messages should be sent to the network operator to obtain additional start information

Whether the VTAM logon monitor facility (referred to as the network solicitor) for local 3270, start-stop, and BSC terminals is to be started when VTAM is started. (See "The Network Solicitor" in Chapter 8 for a description of the network solicitor.)

**Traces:** The trace facility is activated for specified terminals, and the trace continues for as long as the terminals are active or until the network operator stops the trace. See "Starting and Stopping VTAM Facilities" in Chapter 4 for information on stopping the VTAM traces. See "Serviceability Aids" in Chapter 6 for a description of VTAM's trace facility.

**Active Major Nodes in OS/VS and DOS/VS Release 33:** To activate major nodes during VTAM start processing, the network operator can request that VTAM activate the major nodes listed in either a member (or book in DOS/VS) of the VTAM definition library or a VSAM configuration restart data set. The member of the VTAM definition library defines the network's *initial status*; the VSAM data set defines the network's *status prior to deactivation*. If the list is to be taken from the VTAM definition library, the list must be stored under the member name ATCCONxx where xx is a two-character identification created by the user. ATCCON00 is a default member name; other ATCCONxx member names can be specified by the user during start processing. All ATCCONxx members must

be created and filed by the user. If an ATCCONxx member is not created and filed, the default member name ATCCON00 is filed and a warning message is provided when VTAM is started. If a VSAM data set is used, the network operator must have specified that data set on a previous START command to maintain a list of active major nodes. See "Major and Minor Node Structure" earlier in this chapter for a definition of a major node.

**Active Major Nodes in DOS/VS Release 32:** The names of major nodes to be activated when VTAM is started are stored as a book of the VTAM definition library. These names must be stored under the book name ATCCONxx (where xx is a two-character identification number created by the user). ATCCON00 is the default book name; other ATCCONxx book names can be specified by the user during start processing. All ATCCONxx books must be created and filed by the user. If an ATCCONxx book is not created and filed, the default book name ATCCON00 is filed, and a warning message is provided when VTAM is started. To activate major nodes when VTAM is started, the network operator specifies the ATCCONxx name as a start option.

**Active Minor Nodes in OS/VS and DOS/VS Release 33:** The network operator also specifies whether minor nodes are to be activated to their initial status (a *cold restart*) or to their predeactivation status (a *warm restart*). To activate minor nodes to their predeactivation status, VSAM data sets must be defined to record the status of the minor nodes within a major node. Using these facilities, the network operator can control the initial configuration.

**Active Minor Nodes in DOS/VS Release 32:** Minor nodes that are part of an activated major node are activated if they were specified as initially active on their VTAM definition statements. Using these facilities, the network operator can control the initial configuration.

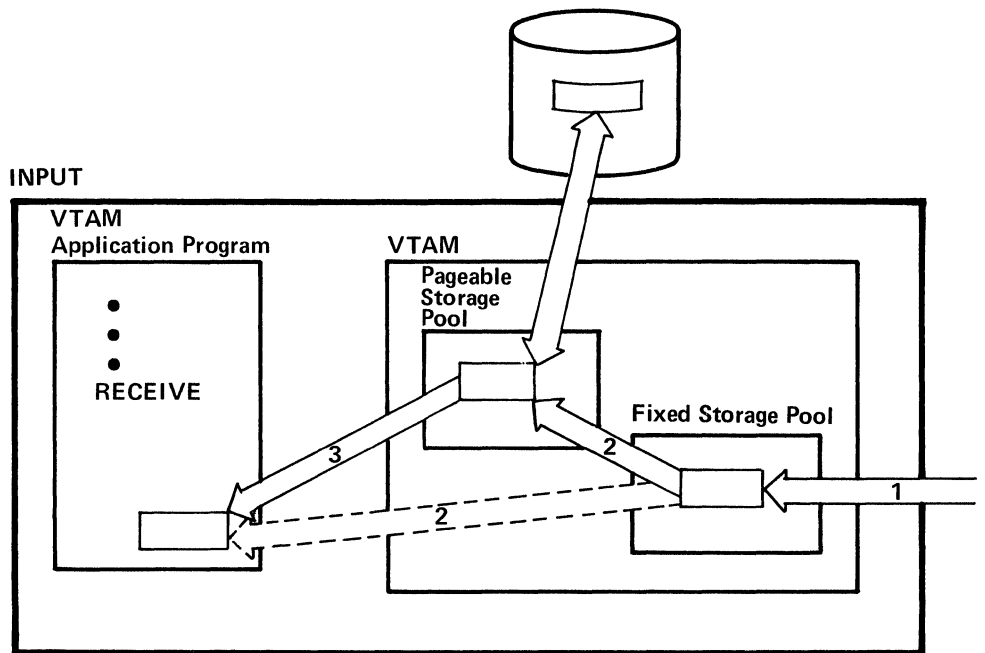
**Storage Pools:** VTAM uses storage pools to allocate space for control blocks, buffers, and channel programs. Pools are established in both fixed and pageable storage. See "Defining VTAM Buffering" later in this chapter for more information on specifying storage pools and on VTAM's use of these pools.

**Prompting Message:** If the network operator is to be prompted, VTAM transmits messages to the system operator's console requesting that start options be entered from the console. The network operator is prompted only (1) if prompting is requested in the ATCSTR00 data set and (for OS/VS) no start parameters are entered with the START command or (2) if an error is encountered by VTAM during VTAM initialization. See "Starting VTAM," in Chapter 4, for information on the role of the network operator in starting VTAM.

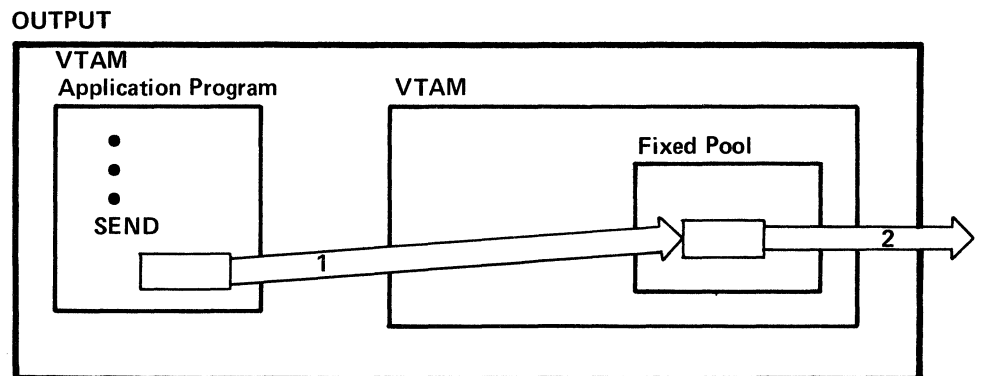
## Defining VTAM Buffering

VTAM is not a queued access method, such as TCAM; VTAM does not save a message until the message is requested by a terminal or an application program. However, VTAM can hold data temporarily until a VTAM application program requests it or until it can be sent to a terminal. Figure 3-10 shows how VTAM buffers input and output data.

VTAM has both fixed and pageable storage pools for buffering data. In addition, storage pools are required for the VTAM control blocks that keep track of incoming data that has not yet been requested by a VTAM application program and to keep track of individual input and output requests. Each storage pool serves all active application programs. Because users' needs vary, VTAM allows the user to specify the size of each of these storage pools. In OS/VS, the user can fix one or more of these pools in main storage. The



- 1 Data arrives from a terminal independent of any VTAM application program request for data. It is placed in a buffer in the fixed storage pool of data buffers.
- 2 If there is an outstanding request for data from the terminal the data is moved from the buffer in the fixed storage pool to the specified VTAM application program data area (shown with dotted lines). If there is no request yet for the data, it is moved from the fixed to the pageable data buffer pool. Later, it can be paged out of main storage.
- 3 When an input request is issued, the data is paged into main storage, if necessary, and moved to the specified VTAM application program data area.



- 1 The VTAM application program requests data transmission. When VTAM has sufficient buffers in the fixed storage pool, it moves the data to the fixed pool.
- 2 The data is sent to the NCP or local terminal, freeing the buffers.

Figure 3-10. How VTAM Buffers Input and Output Data

names of each of the storage pools are different in each operating system and may be found in the appropriate *VTAM System Programmer's Guide*.

When VTAM is started, the user specifies the number of buffers (elements) in each storage pool, the size of each buffer, and for the data storage pool, a threshold number of buffers. The threshold number is the point beyond which, in OS/VS, requests for buffers for input or output are queued until buffers in use become free. Where no value is specified, VTAM supplies a default value. In DOS/VS, VTAM supplies the threshold value; any number specified by the user is ignored. If the threshold is consistently exceeded, program performance and terminal response time may be affected. The user may have to "tune" the buffer number, size, and threshold of its storage pools to achieve an efficient combination of storage utilization and performance.

In addition to specifying the size and threshold of storage pools when starting VTAM, the user can control the amount of buffer storage for data that can be used for each individual connection between an application program and a terminal. Controlling buffer storage for connection between application programs and terminals prevents an individual connection from monopolizing data buffers and slowing down communication between other connections. The number of data buffers for each connection is specified during VTAM definition. VTAM determines the data buffer threshold for each connection by multiplying a number specified for the application program (in the APPL statement) by a number specified for the terminal (in a LOCAL, LU, TERMINAL, COMP, or VTERM statement). For incoming data from terminals, if the threshold is exceeded, VTAM clears any data that may have arrived from the terminal, issues a Clear command if the terminal is a logical unit, and schedules the VTAM application program's LOSTERM exit routine. The program is responsible for resynchronizing the connection. This situation will not occur, however, if the installation ensures that the VTAM application program requests input at a rate that is not below the rate at which data may be coming in to VTAM.

Input data may have to be buffered for a longer time than output data. For input data, VTAM must buffer the data until it is requested by a VTAM application program. For output data, VTAM decides when to begin handling the data; after receiving an output request, VTAM does not move the data from the VTAM application program until it has ensured that it has buffers for it. In general, a VTAM application program should be written so that at least one request for input is always outstanding, or at least so that very little time elapses during which no request for input is outstanding. This may reduce the possibility of exceeding the buffer size for data and control block storage pools.

In OS/VS, the user can specify fixed VTAM storage pools. An installation that requires high performance may want to make resident those pools that would not always be resident because of infrequency of use.

Recommendations and guidelines for determining the sizes and thresholds for storage pools of buffers are provided in the *VTAM System Programmer's Guide* and storage estimates publication for the appropriate operating system.

## CHAPTER 4. CONTROLLING A VTAM SYSTEM

This chapter describes in general how the user controls VTAM using predefined specifications and network operator commands. The appropriate *VTAM System Programmer's Guide* describes predefined specifications in detail and the appropriate *VTAM Network Operating Procedures* describes network operator commands in detail.

### Levels of Control

A user can control VTAM by the specifications made during VTAM definition and by using network operator commands. During VTAM definition, a user defines and tailors a VTAM system. This defines the operating limitations of the VTAM system. Extensive planning is required because the definition of the system and the generated NCPs cannot be easily changed. VTAM network operator facilities allow the network operator<sup>1</sup> to monitor and control a VTAM telecommunication system between the time VTAM is started and the time it is halted.

### VTAM Commands

VTAM's *network operator commands* enable the network operator to monitor and control the telecommunication system. VTAM commands are a subset of the operating system commands. Incorrect commands are rejected by VTAM's command facility, and a message specifying the error is written to the network operator.

Using commands, the network operator (an operator at a system console or an authorized VTAM application program) can:

- Monitor the telecommunications system
- Activate and deactivate nodes
- Initiate requests for connections between terminals and application programs
- Start and stop selected VTAM facilities
- Change line-scheduling specifications for start-stop and BSC lines

In addition, an operator at a system console can start and stop VTAM.

### *Starting VTAM*

VTAM is started by initializing VTAM and the telecommunication network prior to using the VTAM system. Starting VTAM can include the activation of some or all of the nodes. It can also include the activation of selected VTAM facilities.

At the time VTAM is started, the user tailors the telecommunication system either by entering VTAM start options through the network operator's console or by naming a data set that contains predefined start options. The data set is one of the ATCSTRxx members (members for OS/VS, books for DOS/VS) of the VTAM definition library. See "Defining VTAM Start Options" in Chapter 3 for information on predefining VTAM start options in ATCSTRxx data sets.

---

<sup>1</sup>In OS/VS and DOS/VS Release 33, *network operator* refers to either the person at a system console or an authorized application program except when referring to starting VTAM (with the START command) or stopping VTAM (with the HALT command) which can only be done from a system console. See "Network Operator Control from an Authorized Application Program" later in this chapter for a description of issuing operator commands from an authorized VTAM application program. In DOS/VS Release 32, *network operator* refers to the person at the system console.



The VTAM start options that can be entered through the network operator's console are almost the same as those that can be predefined and filed as an ATCSTRxx member. The options that can be entered through the console include:

- An identification number that a physical unit can use to identify the host computer
- Whether VTAM's network solicitor is to be activated
- Whether VTAM's trace facility is to be activated, and if activated, for which nodes
- Which major nodes are to be activated
- Whether minor nodes within an active major node are to be set to their initial status or to the status they had prior to deactivation as defined by their configuration restart data set
- The name of a VSAM data set on which VTAM records configuration restart data
- Sizes of VTAM storage pools and in OS/VS, which pageable storage pools should be fixed in main storage
- The maximum number of major nodes active at one time
- Whether VTAM storage management trace facility is to be activated
- Whether certain messages are to be suppressed
- Whether other start options should be taken from an ATCSTRxx member or book

Except for the last item (a pointer to a list of predefined start options), start options that can be specified by the network operator can also be predefined in an ATCSTRxx member or book.

Starting VTAM differs slightly between DOS/VS and OS/VS. Both are discussed below.

#### **Starting VTAM in DOS/VS**

VTAM is started in DOS/VS by first starting the partition in which VTAM is to be executed and then invoking the VTAM procedure. No VTAM start options can be entered from the console in either of these two steps.

When the network operator starts VTAM, any start options in the ATCSTRxx book are processed first. If prompting was specified in the ATCSTR00 book, VTAM prompts the network operator for start options. If any errors are encountered in the start processing, VTAM prompts the network operator for the last options to be processed before VTAM completes initialization.

#### **Starting VTAM in OS/VS**

The START command starts VTAM in OS/VS. This command can be entered with or without any of the VTAM start options listed above.

When the network operator starts VTAM, any start options in the ATCSTR00 member are processed first. If the operator includes any start options with the START command, these options are processed next. If no start options are specified in the START command and if prompting is specified in the ATCSTR00 member, VTAM prompts the network operator for start options. If any errors are encountered in the start processing, VTAM prompts the network operator for final options. These options are the last to be processed before VTAM completes initialization.

#### **Halting VTAM**

The VTAM HALT command can be used for an orderly, quick, or, in OS/VS1 Release 6 and OS/VS2 MVS, a cancel closedown. Orderly closedown is designed to halt VTAM under normal, planned conditions. Quick and cancel closedown are designated for emergency situations in which halting communications takes precedence over loss of terminal connections and loss of data.

### **Orderly Closedown**

If the network operator specifies an orderly closedown, VTAM does not prohibit connections between terminals and application programs, although new connections of application programs to VTAM are prohibited. VTAM notifies application programs of the pending closedown by scheduling each program's TPEND exit routine. (The TPEND exit routine is an application program exit routine that halts the application program's communications when VTAM is terminating. See "Designing TPEND for the HALT Command" below for information on the TPEND exit routine.)

Except for prohibiting the opening of access method control blocks (ACBs), VTAM allows normal operation until all application programs have closed their ACBs. Then VTAM deactivates all nodes and closes down the telecommunication system.

For VTAM's orderly closedown, the user should ensure that each application program has a TPEND exit routine and that each TPEND exit routine is designed to halt communications in an orderly manner and then disconnect the program from VTAM. If an application program connected to VTAM does not have a TPEND exit routine, it is not notified of the pending closedown. This delays the halt either until the application program closes its ACB or the network operator cancels the application program.

### **Quick Closedown**

If the network operator specifies a quick closedown, VTAM prohibits any further communication between terminals and application programs. New connections of application programs to VTAM are also prevented. No additional input or output requests are accepted, although data already read into VTAM buffers from local 3270, BSC, and start-stop terminals can be read by application programs. VTAM also notifies application programs of the pending closedown by scheduling each program's TPEND exit routine. Having scheduled each exit routine, VTAM waits until all application programs have closed their ACBs before it closes down the telecommunication system.

So that the quick closedown functions properly, the user should ensure that each application program using VTAM has a TPEND exit routine. Because a quick closedown is probably indicative of an emergency situation, the exit routine should do no more than initiate a closedown procedure.

### **Cancel Closedown (OS/VS1 Release 6 and OS/VS2 MVS Only)**

In OS/VS1 Release 6 and in OS/VS2 MVS, the network operator can cause immediate closedown of VTAM using the HALT command with the CANCEL option. VTAM notifies application programs of the shutdown by scheduling each program's TPEND exit routine or abnormally terminating the programs that do not have a TPEND exit routine. No further input or output operations are performed, and VTAM becomes immediately inactive. Local devices are released and storage used by VTAM is released. The CANCEL option is intended for emergency use only. The ability to cancel VTAM using the HALT command is not available in DOS/VS, OS/VS2 SVS, or OS/VS1 Release 5 or prior releases.

### **Designing TPEND for the HALT Command**

Each application program's TPEND exit routine is scheduled whenever VTAM is about to terminate. If VTAM is terminating as the result of a HALT command, input to the exit routine indicates whether the closedown is orderly, quick, or cancel.

For an orderly closedown, an application program can complete current processing, notify connected terminals that the telecommunication system is closing down, and disconnect itself from VTAM. For a quick closedown, the application program should do no more than close its ACB. Note that the ACB cannot be closed in the TPEND routine.

For quick closedown, if all application programs do not close their ACBs within 45 seconds, VTAM notifies the network operator indicating the names of the open ACBs. The network operator can then either permit the application programs to continue

processing and wait until the ACBs are closed or can use the facilities of the operating system to cancel the job containing the application programs.

For cancel shutdown, the application program can make no VTAM requests except to issue a CLOSE macro instruction. VTAM is closed down whether or not all application programs close their ACBs.

Because a CLOSE macro instruction cannot be issued in an exit routine, this routine should set an indicator that alerts the main portion of the application program to the pending shutdown. In the case of a quick shutdown, for example, the shutdown procedure could post an event control block (ECB) and return to VTAM. This ECB should have been previously created by the main portion of the application program, and the main portion of the program should have been waiting for it to be posted. When the ECB is posted, the application program regains control and can then issue a CLOSE macro instruction for its VTAM ACB. The CLOSE request automatically disconnects any terminals connected to the application program.

### **Monitoring VTAM Status**

The network operator monitors the VTAM system by requesting and studying status information for nodes in the system. VTAM's DISPLAY command enables the network operator to request status information and to verify changes resulting from previous operator requests. This command enables the operator to request information about the following types of nodes:

- Application programs

- Terminals (logical units and local 3270, BSC, and start-stop terminals)

- Communication lines

- BSC and start-stop switched-line ports. (Status displays for ports are the same as for telecommunication lines.)

- Physical units

- Network control programs (NCPs)

To display the status of a node, the network operator specifies the symbolic name of the node in the DISPLAY command. A minor node specified in a VTAM DISPLAY command must be part of an active major node. An NCP major node specified in a DISPLAY command must itself be active. The information displayed by VTAM for each type of node includes:

**For an Application Program:** Whether the application program is currently connected to VTAM, the names of terminals connected to the application program, and the names of terminals queued for logon to this application program. Because an open ACB is an application program to VTAM, a program can be executed in the system but not recognized by VTAM as an application program if it does not have an open ACB for VTAM. Displays can still be requested of application program minor nodes for which there is currently no open ACB; such a display indicates that the application program is inactive (not connected to VTAM).

**For Terminals:** Whether the terminal is active or inactive, the name of the application program (if any) to which the terminal is allocated (that is, connected or queued for logon), the name of the application program (if any) for which an automatic logon is specified for the terminal, whether or not a logical unit is in a state of being connected or disconnected, the name of the switched major node if a logical unit is part of a switched major node, the name of the local major node if a logical unit is part of a local major node and the channel unit address of the associated physical unit, a list of traces in effect for the terminal, the current specification of the device transmission limit (for polled

start-stop and BSC terminals only), the device type, a count of the input/output activity and temporary errors for the terminal, and the names of the line-group and the line to which the terminal is assigned (for remote terminals) or the channel and unit address (for a local 3270).

**For Communication Lines:** Whether the line is active or inactive; the name of the group to which the line is assigned; the names of all nodes assigned to the line and an indication of those that are active; the current specifications for polling delay, negative response to polling limit, and NCP session limit (for polled start-stop and BSC lines only); whether the line is switched or nonswitched; and whether a line trace is active for the line. In addition, if the line is SDLC, VTAM displays the names of the SDLC cluster controllers that are assigned to the line and indicates those that are active, or displays the names of the remote communications controllers that are assigned to the line and indicates those that are active.

**For a Physical Unit:** Whether the unit is active or inactive, whether a trace is active for the unit, the names of the VTAM terminals assigned to the control unit and an indication of those that are active, the names of the line and of the line-group to which the unit is assigned, and path information for physical units on switched lines.

**For NCPs:** Whether the NCP is active or inactive, the channel and unit address and the type of the communications controller containing the NCP (if the NCP is for a local communications controller), a list of traces in effect for the NCP the load-module name of the NCP, a count of the input/output activity and of temporary errors for the communications controller in which the NCP resides (if the NCP is for a local communications controller), and an indication of whether the NCP is in a local or remote communications controller.

### ***Activating and Deactivating Nodes***

Before a node can be used in a VTAM system, the node must be active. The user can control the activation and deactivation of many of the nodes in VTAM, including both major and minor nodes. All major nodes can be explicitly activated and deactivated by the network operator, including:

- NCPs for local and remote communications controllers
- Sets of physical units and logical units on switched lines
- Sets of local physical units and logical units
- Sets of local 3270s
- Sets of application programs

When VTAM is started, major nodes can be activated by using VTAM start options; all active major nodes are deactivated when VTAM is halted. Major nodes can also be activated and deactivated with VTAM's VARY command while VTAM is being executed.

To activate or deactivate a major node after VTAM is started, the network operator enters a VARY command that contains the name of the node and indicates whether the request is for activation or deactivation. For an activation request, the node name is the name of the member or book of the VTAM definition library that contains the definition statements for the node. For a deactivation request, the name is the name that was given in an activation request.

See "The Major and Minor Node Structure" in Chapter 3, for a definition of major nodes. See "Starting VTAM" and "Halting VTAM," earlier in this chapter, for information on activating and deactivating nodes by other than the use of the VARY command.

Once a major node has been activated, some minor nodes within it can be activated and deactivated by the network operator while VTAM is running. The minor nodes that can be dynamically activated or deactivated at the request of the network operator are:

- SDLC, start-stop, and BSC lines
- Ports
- Physical units
- Logical units and local 3270, BSC, and start-stop terminals
- BSC and start-stop terminal components

In OS/VS and DOS/VS Release 33, the user can also specifically request the activation of any of these minor nodes when VTAM is started by:

- Specifying in a minor node's definition statement that the node is to be activated when its major node is activated and then specifying the COLD option on the START command. If the start options indicate that the major node is to be initially active, all associated minor nodes defined as active are automatically activated.
- Defining a configuration restart data set for a major node and then specifying the WARM option on the START command. If the start options indicate that the major node is to be initially active, all associated minor nodes that were active when the major node was deactivated will be reactivated.

In DOS/VS Release 32, the user can also specifically request the activation of any of these minor nodes when VTAM is started (except for communication lines). A minor node's definition statement can specify that the node is to be activated automatically when its major node is activated; VTAM start options activate that major node. SDLC, BSC, and start-stop lines are automatically activated by VTAM when an NCP is initially activated and can be activated and deactivated thereafter by network operator commands.

As noted earlier in this chapter, halting VTAM (by using VTAM's HALT command) automatically deactivates all active nodes. The orderly mode of VTAM's HALT command deactivates the nodes only when all application programs have closed their ACBs. The quick mode of the command begins to deactivate all nodes after the last application program has been notified of the pending halt, although VTAM termination is not completed until all ACBs have been closed.

To activate or deactivate a minor node while VTAM is being executed, the network operator must enter a VARY command containing the name of the node and indicating whether the request is for activation or deactivation.

**Note:** *A major node containing the minor node definition must be active before that minor node can be activated or deactivated.*

The VARY command can be used for normal deactivation, immediate deactivation, forced deactivation, and restart deactivation. When normal deactivation is specified, the node is not actually deactivated by VTAM until the associated application program and terminal connections have been terminated by either the application program or by the terminal operator. Queued requests for connections involving the nodes in the deactivation are dequeued. No new requests for connection with these nodes are accepted. An application program connected to a terminal to be deactivated is not notified of pending normal deactivations.

Two common uses for normal deactivation are:

Allowing many terminals to use a network by periodically and temporarily making some terminals inactive

Closing down a portion of the total network of terminals, perhaps as a routine end-of-the-day operation

For a large network of terminals, normal deactivation can be used to deactivate some terminals as they log off from a VTAM application program, allowing other terminals to be activated or allowing the remaining active terminals to be responded to more quickly. Later, as network traffic decreases or as the deactivated terminals are to be given another turn, they can be reactivated and possibly other terminals can be normally deactivated. This use of normal deactivation should be defined by the system programmer and understood by the network operator and possibly by terminal operators who may wish to know why they are temporarily being denied access to the network.

Normal deactivation can be used as a routine means of ensuring that some terminals, as soon as they have logged off, are deactivated and thus unable to use the network for the remainder of the day. Meanwhile, the resources of the network can be used for other terminals that are to be activated or to remain active (perhaps in a different time zone). This use of normal deactivation also requires coordination between the network operator, VTAM application programs, and possibly terminal operators. Since a VTAM application program is not directly informed when the network operator specifies normal deactivation of a terminal to which it is connected, other means may be required to notify the program to send a message to a terminal that is logging off that it will soon be deactivated and unable to request logon to another VTAM application program. This means could be program-to-network operator communication (a WTOR macro instruction) or a time-of-day routine.

When immediate deactivation is specified, all queued requests for connections to nodes included in an immediate deactivation request are dequeued; no new requests for connection with these nodes are accepted. All input or output operations for these nodes are immediately halted, with possible loss of data. (For local 3270, BSC, and start-stop terminals, data that is already in VTAM buffers prior to the deactivation can still be obtained by the application program that was connected to the terminal; data in transit to VTAM from the terminal may be lost.) If the deactivation involves a terminal connected to an application program, the application program is notified of the deactivation by the scheduling of the program's LOSTERM exit routine; the application program must disconnect the terminal for the deactivation to be completed. (See "Application Program Concepts and Facilities" in Chapter 5 for details on the LOSTERM exit routine.)

The immediate mode provides very tight control over the network; normal mode provides less stringent control, but allows for a more orderly deactivation. For deactivation to be completed, both modes require that application programs disconnect terminals included in the deactivation.

Normal or immediate deactivation is not completed until all connections between application programs and affected terminals are broken and the proper commands and responses are exchanged. If conditions in the network prevent the sending of the proper commands or responses, normal and immediate deactivation are not successful. Forced deactivation and restart deactivation are used when immediate and normal deactivation are not successful. *Forced deactivation* causes VTAM to deactivate the node internally, without waiting for the responses to deactivation commands or for application programs to disconnect affected terminal. *Restart deactivation* causes a forced deactivation and then a warm restart.

## Starting and Stopping Application Programs

Starting a VTAM application program requires that the network operator activate the major node containing the APPL definition statement of the application program and start the job containing the application program. The major node can be activated explicitly by using the VARY command or implicitly, when VTAM is started, by using start options. The application program job is started like any other job in the system. The order of the two operations is not critical; however, the major node must be active when the application program attempts to open its ACB for VTAM.

The major node containing the application program definition must remain active as long as the application program's ACB is open. If an attempt is made to deactivate a major node with the VARY command and all associated ACBs have not been closed, the command is rejected.

To VTAM, an application program is one that is defined within an active major node and that has an open ACB for VTAM. Thus, for VTAM, stopping an application program requires only that the ACB be closed. But the major node itself cannot be deactivated until all application program's minor nodes have closed their ACBs.

An ACB is closed when the application program issues a CLOSE macro instruction. It is also closed by the operating system when the application program is terminated. An application program major node is deactivated by the VARY command and by the HALT command.

## Activating and Deactivating Local 3270s

Activating a minor node for a local 3270 allocates that terminal to VTAM if it is available. If that terminal is not available (that is, if it is allocated to another, non-VTAM user), the activation request is rejected.

In OS/VS and DOS/VS Release 33, activating a major node for a set of local 3270s with the COLD option activates those terminals that are available and are defined in the associated LOCAL definition statements as initially active. Activating a major node for a set of local 3270s with the WARM option activates those terminals that are available and were active when the major node was deactivated. VTAM notifies the network operator of any terminals that are not available. The network operator can then use the VARY command to activate the minor nodes for these terminals as they become available.

In DOS/VS Release 32, activating a major node for a set of local 3270s allocates those terminals to VTAM that are available and are defined in the associated LOCAL definition statements as initially active. VTAM notifies the network operator of any terminals that are not available. The network operator can then use the VARY command to activate the minor nodes for these terminals as they become available.

Deactivating a minor node for a locally attached 3270 returns the terminal to the operating system. Deactivating the major node for a set of local 3270s returns all active terminals in that set to the operating system. (Inactive terminals in that set are not currently allocated to VTAM.)

If an automatic logon is specified for a local 3270, a logon is queued to the application program (if it is active and accepting logons) whenever the terminal is activated. If the application program has an active LOGON exit routine, the routine is scheduled.

When a local 3270 is activated, it is available for connection to application programs using VTAM. When the terminal is deactivated, it is unavailable for connection through VTAM, but it is available to non-VTAM users.

### Activating and Deactivating Local SNA Major Nodes

A local SNA major node can be controlled by activating and deactivating the entire set of its minor nodes or individual minor nodes (its physical units and logical units). The VARY command can control the entire major node by specifying the name on the local major node definition statement (VBUILD). An individual minor node is controlled by specifying the name on a PU or LU statement.

### Activating and Deactivating an NCP

Activating an NCP for a local communications controller allocates the controller to VTAM. Until the NCP is deactivated, the communications controller remains allocated to VTAM and is not available to other users of the operating system except through VTAM facilities. Allocating a local communications controller to VTAM also implicitly allocates the associated remote attachments to VTAM. (Only local devices are recognized by the operating system and need to be explicitly allocated to VTAM; the remote devices are "allocated" to VTAM because they are part of the network controlled by the local communications controllers.)

**In OS/VS and DOS/VS Release 33**, activating an NCP can also initiate the loading of the NCP into the controller. The NCP is loaded if the proper NCP is not in the controller or the network operator specifically requests reloading. When the network operator activates the NCP, WARM is specified to reactivate the minor nodes within the NCP major node to their status prior to deactivation or failure, or COLD is specified to activate the minor nodes as specified by their definition statements.

**In DOS/VS Release 32**, activating an NCP can also initiate the loading of the NCP into the controller. The NCP is not loaded if the NCP specified in the activate command is already loaded and is unmodified by the network operator in its initial state. If the specified NCP is not in the communications controller, it is loaded. If a currently loaded NCP is to be used, it must be in its initial state; that is, its status is that as specified for the NCP in the VTAM definition library. Thus, active physical units, logical units, cluster control units, and local 3270, BSC, and start-stop terminals and components are only those that are specified as active in the definition statements for the NCP. If an NCP is modified by the network operator, it is not considered to be in its initial state.

VTAM does not automatically reestablish connections between application programs and terminals, although automatic logons are initiated for active terminals that have automatic logon specifications. Connections must be reestablished by using the OPNDST macro instruction. Also, terminals attached to switched lines have been disconnected and must be redialed.

Deactivating an NCP for a local communications controller returns the communications controller to the operating system. It does not delete the loaded NCP. The user is responsible for loading another control program if the current one is not acceptable for the next user.

Terminals attached to a remote communications controller are not available for use by VTAM until the remote controller has been activated. To activate the remote communications controller, both NCPs must be activated: first the NCP for the local communications controller and then the NCP for the remote communications controller. Before deactivating a local communications controller in a VTAM system, the network operator does not have to deactivate remote controllers. Deactivating an NCP for a local communications controller automatically deactivates any remote communications controllers attached to it.

**Note:** *Activating an NCP for a remote communications controller requires the line connecting the remote controller to the local controller to be active.*



If the NCP contains PEP, activation and deactivation requests may have an impact on emulation processing. Activating an NCP with PEP can cause the entire NCP to be loaded if necessary (including both the emulation functions and the network control functions), although deactivating the NCP through the VARY command makes the NCP and the communications controller inactive only for VTAM. Deactivation does not halt emulation processing in the controller.

**In OS/VS and DOS/VS Release 33**, VTAM's activation and deactivation requests for communication lines may also affect emulation mode. VTAM controls the assignment of lines that can be reassigned between network control and emulation modes. The network operator can assign a line to emulation mode or network control mode as initially defined to VTAM (a cold start) or as defined by its major node's configuration restart data set (a warm start). When they are deactivated, the lines are returned to emulation mode. See "Network Control Program Requirements" in Chapter 7 for more information on controlling an NCP with PEP.

**In DOS/VS Release 32**, VTAM's activation and deactivation requests for communication lines may also have some impact on emulation mode. VTAM controls the assignment of lines that can be reassigned between network control and emulation modes. Lines that can be reassigned are automatically assigned to emulation mode until they are activated by VTAM. When they are activated, lines that can be reassigned but are currently assigned to emulation mode are reassigned to network control mode (except those lines being used by emulation mode). When they are deactivated, the lines are returned to emulation mode. See "Network Control Program Requirements" in Chapter 7 for more information on controlling an NCP with PEP.

#### **Activating and Deactivating Remote Attachments**

Activating and deactivating a minor node for a remote attachment (such as a terminal, physical unit, or line) does not affect the allocation of that attachment to VTAM. Remote attachments remain implicitly allocated to VTAM as long as the NCP for the local communications controller is not deactivated by VTAM.

**In OS/VS and DOS/VS Release 33**, for VTAM to treat a terminal as active (that is, for VTAM to permit a terminal to be connected to or queued for connection to an application program), the associated line, physical unit, and terminal must all be active. When the NCP is initially loaded, lines, physical units, logical units, and BSC and start-stop terminals and components are activated as specified in the definition statements for those minor nodes or as specified by the NCP's configuration restart data set, depending on whether the network operator enters the WARM or COLD option.

**In DOS/VS Release 32**, for VTAM to treat a terminal as active (that is, for VTAM to permit a terminal to be connected to or queued for connection to an application program) the associated line, physical unit, and terminal must all be active. When the NCP is initially loaded, physical units, logical units, and local 3270, BSC, and start-stop terminals and components are activated as specified in the definition statements for those minor nodes. SDLC, start-stop, and BSC lines are automatically activated when an NCP is initially loaded.

Active application programs can be connected only to active terminals (logical units or local 3270, BSC, or start-stop terminals). Activating and deactivating application programs and local terminals are discussed above. Activating and deactivating remote terminals are discussed below.

Although an application program can be connected to an active terminal, the connection can be completed only if the terminal is accessible through an active line, an active NCP, and, for an SNA terminal, an active physical unit. (If the terminal is accessible, as defined in its TERMINAL or VTERM definition statement, through more than one line, at least

one of the lines must be active for connections to be completed.) Thus, a remote terminal is treated as active (that is, connected to or available for connection to an application program) only if all nodes in a valid path to the terminal have been activated.

Deactivating the NCP, the line (or lines in the case of switched networks), the physical unit (for SNA terminals), or the terminal (logical unit, or BSC or start-stop terminal) effectively deactivates the terminal by making it unavailable for connection to any program using VTAM. Each of these minor nodes is deactivated by using VTAM's VARY command with the deactivate option. To reactivate a terminal, the VARY command with the activate option must be issued for the minor node that was deactivated.

Activation and deactivation are essentially the same for all remote terminals, whether they are attached to a local communications controller or to a remote communications controller. The only difference is that a terminal attached to a remote communications controller depends on the status of a greater number of nodes to complete an active path to an application program. In addition to the status of the NCP of the local communications controller, a terminal attached to a remote controller depends on the status of the following nodes associated with that remote controller:

The remote communications controller itself

The line or lines and, for all SNA terminals, the physical unit connecting the terminal to the remote controller

#### **Example of Deactivating and Reactivating a Remote Logical Unit**

Assume an active logical unit is connected to an active physical unit which is in turn accessible on an active nonswitched line. The logical unit can be deactivated by using the VARY command to deactivate any of the following:

The NCP

The line

The physical unit

The logical unit

Reactivating the logical unit depends on the node that is specified in the deactivation request. To reactivate the logical unit, an activation request specifying the same node must be entered so that, for example, if the NCP was specified in the deactivation request, the NCP is activated.

**In OS/VS and DOS/VS Release 33**, if a physical unit or logical unit is effectively deactivated by deactivating the NCP, it can be reactivated automatically when the NCP is reactivated if (1) a cold restart is specified and they are defined by VTAM definition statements as being initially active or (2) if a warm restart is specified and a configuration restart data set exists for the NCP major node. A physical unit or logical unit that cannot be automatically reactivated must be explicitly reactivated.

**In DOS/VS Release 32**, if a physical unit or logical unit is effectively deactivated by deactivating the NCP, the physical unit or logical unit can only be reactivated automatically when the NCP is reactivated if they are defined by VTAM definition statements as being initially active when the NCP is loaded. A physical unit or logical unit not automatically activated must be explicitly reactivated.

#### **Activating and Deactivating Switched SNA Major Nodes**

Switched SNA major nodes can be activated or deactivated by using the VARY command for the entire switched major node or portions of the switched major node. The VARY command can control the entire switched major node by specifying the name on the switched major node definition statement (VBUILD). The portions of a switched major node that use the dial-out facility can be controlled by activating and deactivating an

individual path by specifying the path identifier (PID), or a group of paths by specifying the group identifier (GID). For example, all paths that use WATS lines for dial-out operations can be associated with a GID and activated or deactivated as a unit.

Dial-in operations can be controlled by activating or deactivating a line's ability to answer a dial-in request. Lines used for dial-in operation can be activated or deactivated; lines used for dial-in or dial-out operation can be changed to dial-out only.

For additional information on controlling switched major nodes, see "Network Control Program Requirements" in Chapter 7.

### Special Considerations for Activation

When VTAM activates a major node, it builds a segment of a table containing entries that describe the minor nodes defined for that major node. (When the major node is deactivated, the table segment for that node is deleted from main storage.) VTAM uses the entries in the table to represent the minor nodes. It is these entries that VTAM allocates to users; VTAM does not allocate the node the entry represents, but it retains the ownership of all the nodes. In most instances, the status of the minor node and of its representative table entry is the same; thus, no distinction is usually necessary between a node and its table entry. The activation of application programs and the activation of local 3270, BSC, and start-stop terminals are exceptions.

**An Active Application Program:** VTAM activates only major nodes for application programs. Once an application program major node is active, each table entry for a minor node within it is available to form a connection between VTAM and an application program. This connection is made when an ACB (pointing to one of these entries) is successfully opened. A table entry for an application program minor node can be used to form a connection with VTAM by only one application program at a time; that is, only one open ACB at a time can refer to it.

VTAM's VARY command and the start options for activating nodes activate only the major nodes for application programs. The user must start the application program and open the ACB. When the ACB is open, VTAM treats it and its associated table entry as an active application program.

**An Active Terminal:** When VTAM activates a remote logical unit, it indicates in the logical unit's definition table entry that it is active and transmits an activation command to the NCP. For logical units on switched lines, the activation command is not sent until a dial-in or dial-out operation is performed. When VTAM activates a local logical unit, it indicates in the logical unit's definition table entry that it is active and transmits an activation command to the associated physical unit after connection is initiated. When VTAM activates a local 3270, BSC, or start-stop terminal, it indicates in the terminal's definition table entry that it is active; no activation command is sent for local 3270, BSC, or start-stop terminals.

For start-stop and BSC terminals, the physical status of the terminal can differ from that of its table entry; for example, a table entry can be marked active even though the terminal it represents has not been turned on or is not even physically in the network. But, because VTAM allocates the table entry to the application program when completing a connection request, an application program can connect to a terminal that is not physically part of the network. Connection is possible in this case if the table entry is marked active and a defined path has been activated.

If connection is requested to a nonexistent (but defined) terminal, the connection is completed, but no data transfer can occur. These conditions enable the user to include, in the definition of the network, resources that are not physically available but will be

added at a later date. By doing this, the user can avoid redefining the network or recoding application programs to add minor nodes to the telecommunication system.

For logical units on nonswitched lines, the status of the table entry and the terminal it represents must agree. Thus, when a logical unit is activated, it must be physically active before its table entry can be activated and made available for use by an application program. (A terminal without an active path is still not available for connection to an application program; in effect, the terminal is inactive for application programs.)

### ***Initiating Requests for Connection***

In addition to its use in activating and deactivating nodes, VTAM's VARY command can initiate connection on behalf of terminals. To initiate a connection, the network operator enters a VARY command containing the names of the terminal and the application program to be connected. VTAM then processes the command as though it were an automatic logon for the terminal; VTAM notifies the application program that a logon request was received from a terminal, but the application program must then accept the request before connection is completed. (See "Application Program Concepts and Facilities" in Chapter 5 for details on connecting application programs and terminals.)

In addition to initiating a logon, this option of the VARY command alters the automatic logon specification for the terminal. For example: If the network operator initiates a logon on behalf of terminal 1 for application program A, terminal 1 is initially logged on to that program. Thereafter, whenever terminal 1 is available, it is automatically logged back on to application program A unless program A releases it. This automatic logon specification for terminal 1 remains in effect until the network operator either enters a new logon on behalf of the terminal or deactivates the major node containing the terminal definition and reactivates the major node with the COLD option. Because the network operator logon modifies the automatic logon specification, care should be exercised when using the logon facility of the VARY command.

### ***Starting and Stopping VTAM Facilities***

Some of the facilities in VTAM need not be active continuously. VTAM allows the network operator to start and stop these facilities selectively. Using VTAM's MODIFY command, the network operator can start the following VTAM facilities:

- Network solicitor, a facility for local 3270, BSC, and start-stop terminals, described in Chapter 8

- Trace facility, described in Chapter 6

- Print-trace utility program, described in Chapter 6

- Dump utility program, described in Chapter 6

- Teleprocessing Online Test Executive Program (TOLTEP), described in Chapter 6

The MODIFY command can be used to stop the network solicitor and the trace facilities. The network solicitor and the trace facilities can also be activated by VTAM start options. See "Starting VTAM," earlier in this chapter, for more information on activating these two facilities at start time.

### ***Suppressing Network Operator Messages***

VTAM allows the user to suppress certain classes of network operator messages. Suppression can be specified when starting VTAM, either as a predefined start parameter or by the operator, or when operating VTAM by using the network operator MODIFY command. Suppressing messages allows the user to adjust the amount of information it receives from VTAM to suit the needs of the operator.

Messages are divided into these classes:

- Information
- Warning
- Serious
- Normal

Messages that prompt information from the operator and messages that are responses to a DISPLAY command cannot be suppressed.

### ***Changing Line-Scheduling Specifications***

VTAM permits the network operator to change the following line-scheduling specifications for polled, nonswitched start-stop or BSC lines:

- Polling delay
- Negative response to polling limit
- NCP session limit
- Device transmission limit

The first three specifications are made on the NCP's LINE statement. The last specification is made on the NCP's TERMINAL statement. Refer to the *NCP Generation* publication for a description of these parameters and their functions.

**In OS/VS and DOS/VS Release 33**, the line-scheduling specifications are changed by the network operator by using an option of VTAM's MODIFY command. A line-scheduling change remains in effect until the network operator changes the specification again using the MODIFY command or the communications controller is deactivated and reactivated with the cold option.

**In DOS/VS Release 32**, the line-scheduling specifications are changed by using an option of VTAM's MODIFY command. A line-scheduling change remains in effect until the network operator changes the specification again using the MODIFY command or the communications controller is deactivated. When the communications controller is reactivated, the line-scheduling parameters specified when the NCP was generated are used.

## **Network Operator Control from an Authorized Application Program**

This section gives a general description of a *program operator*, that is, a VTAM application program used as a network operator. This section applies to only OS/VS and DOS/VS Release 33; the program operator capability is not available in DOS/VS Release 32. A VTAM application program can be authorized to use the SENDCMD macro instruction to issue VTAM network operator commands (except START and HALT) and the REPLY command, and the RVCMD macro instruction to receive VTAM network operator messages. The REPLY command has the same format and meaning as a REPLY command entered from an OS/VS system console. For DOS/VS users, this command is only used by a VTAM application program to respond to a prompting message from VTAM. This facility can allow a user to:

- Enter network operator commands from a terminal in the network
- Monitor and control elements in the network at program execution speed
- Define specialized commands (for example, to display the status of the entire network)
- Reformat responses to VTAM commands (such as to reformat the status display of the entire network to fit on a 3270 display screen)

When defining a program operator to VTAM, a user may specify:

**AUTH=PPO** to specify that the program operator is authorized to use the SENDCMD macro instruction to enter VARY, DISPLAY, MODIFY, and REPLY commands and the RCVCMD macro instruction to receive *both* solicited (in response to a network operator command entered by the program) and unsolicited VTAM messages.

**AUTH=SPO** to specify that the program operator is authorized to use the SENDCMD macro instruction to enter VARY, DISPLAY, MODIFY, and REPLY commands, and the RCVCMD macro instruction to receive *only* solicited VTAM messages.

One program operator with AUTH=PPO can be active at one time and any number of program operators with AUTH=SPO can be active at one time. However, any number of each type may be defined.

In addition to the SENDCMD and RCVCMD macro instructions, a program operator can use any of the facilities available to other VTAM application programs as described in Chapters 5 and 8. A program operator can be written to allow network operator control from terminals in the network by using VTAM connection and communication macro instructions to communicate with terminals in the network and using the SENDCMD and RCVCMD macro instructions to communicate with VTAM network operator facilities. Figure 4-1 illustrates the communication paths used by such an application program.

To enter an operator command, a program operator issues a SENDCMD macro instruction specifying a request parameter list (RPL) that contains the address of a data area. The data area contains a network operator command in the same format and with the same meaning as a command entered from the system console, preceded by a header that is used to correlate responses with the entered command and to request a response from VTAM.

To receive an operator message from VTAM, a program operator issues a RCVCMD macro instruction specifying a request parameter list (RPL) that contains the address of a data area. Upon completion of the request, this area contains the VTAM message in the same format and with the same meaning as a message received at a system console. The message is preceded by a header that is used to correlate messages with entered commands and to distinguish between solicited and unsolicited VTAM messages.

*VTAM Macro Language Reference* has a detailed description of the SENDCMD and RCVCMD macro instructions. *Supplement to the VTAM Macro Language Guide for the Program Operator*, describes this facility and has a sample program that uses the SENDCMD and RCVCMD macro instructions to allow network operator control from terminals in the VTAM network.

## Considerations for Network Operator Control

To use VTAM's network operator facilities effectively, the user must establish procedures for monitoring and controlling the VTAM system. These procedures must be defined for operators that enter network operator commands (either from a system console or from a terminal connected to a program operator) and for programmers who write program operator application programs. Considerations for a program operator are similar to those for an operator at a console or terminal. Planning for terminal and console operators should include:

*Names of major and minor nodes and data sets:* Because VTAM manipulates symbolic node names, the operator must know the names of all major and minor nodes that are to be used in VTAM commands or that can be received in VTAM messages. The operator should also know the names and uses of all VTAM data sets. (See "VTAM Data Sets" in Chapter 7 for a description of the data sets used by VTAM.)

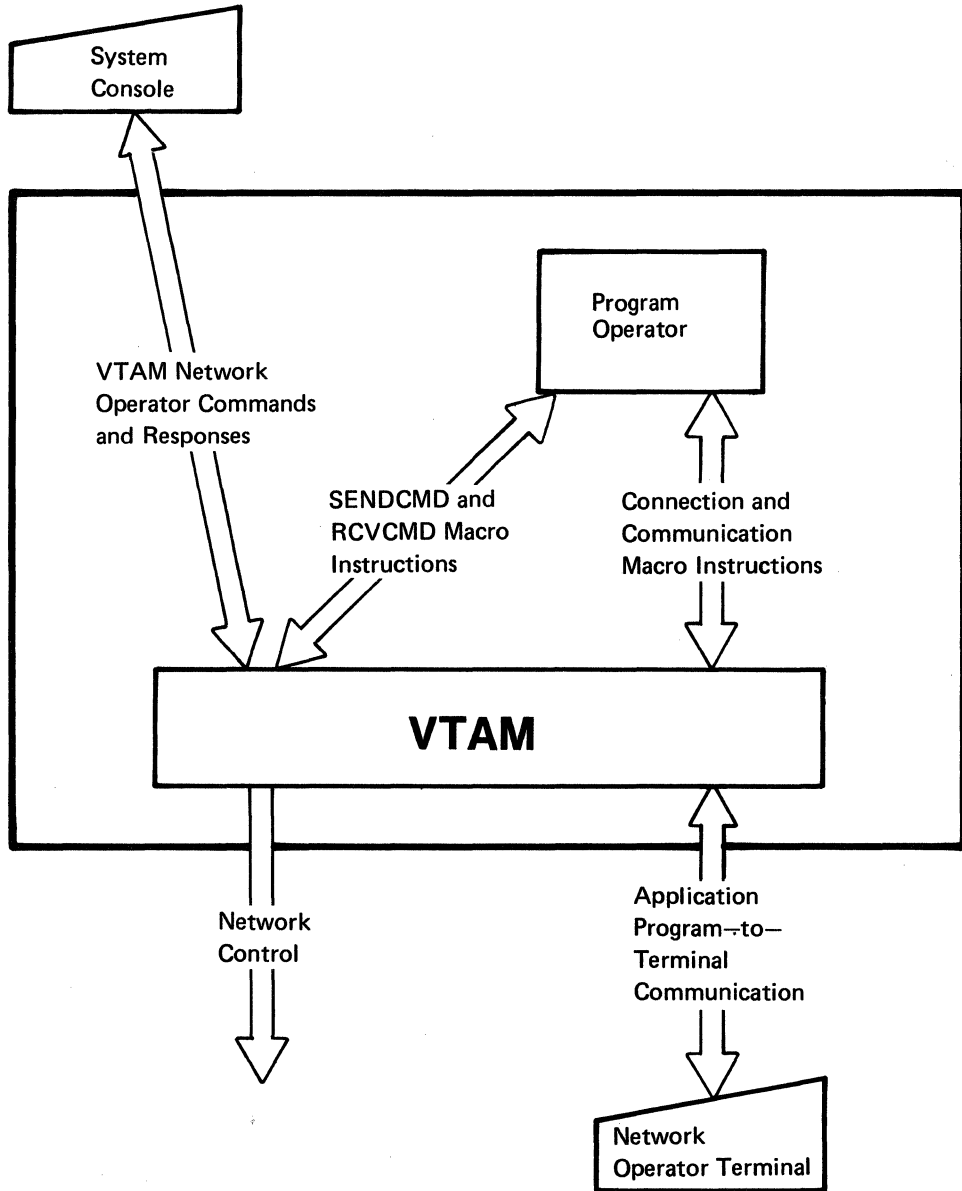


Figure 4-1. Network Operator Control of the VTAM System

*Relationship between the names of VTAM application programs and job names:* Application programs are identified to VTAM through APPL statements and ACBs. For VTAM, the name of an application program is the name of an APPL statement; for the operating system, the job name or the step name is the program name. The operator needs to know how each application program name in VTAM relates to job names or step names.

*Relationship between the symbolic and the physical network:* Because VTAM commands function differently for major nodes than for minor nodes, the network operator should know which nodes are major and which are minor. Also needed is a knowledge of the hierarchical structure of the NCP nodes and of the relationship between the symbolic names and the physical units they represent.

*Impact of VTAM on emulation in an NCP with PEP:* Although VTAM does not use emulation mode, the access method can have an impact on the operation of emulation mode in an NCP with PEP. (See “Network Control Program Requirements” in Chapter 7 for a description of VTAM’s impact on emulation mode.)

*Switched network support:* Controlling a switched network requires an understanding of how such a network is defined. See “Network Control Program Requirements” in Chapter 7 for details on defining and controlling switched lines and terminals that use them.

*Storage sizes:* When the operating system is loaded, the VTAM region or partition size can be selected. VTAM also allows the system console operator to make storage pool specifications. If these two specifications are not predefined to the system or to VTAM, the operator needs the detailed storage information.

*Procedures:* The user may also want to establish procedures for the operator to follow. The procedures can be for both planned normal operations and contingencies. The procedures might cover: when and how to start and stop VTAM, application programs, and the network solicitor; when, how, and which terminals and other nodes to activate and deactivate; and what actions to take when network errors are encountered, including what traces to activate, and when and how to activate, deactivate, and dump an NCP.





## CHAPTER 5. VTAM APPLICATION PROGRAMS

This chapter introduces VTAM application program concepts and describes the considerations involved in writing a VTAM application program. The first section summarizes the VTAM macro instructions. The second section explains the organization of a VTAM application program. The third section explains the use of the VTAM language. The reader planning to write VTAM application programs may wish to go directly to *VTAM Macro Language Guide* rather than reading this chapter to become acquainted with VTAM application programs. *VTAM Macro Language Guide* expands upon the information provided in this chapter.

Most of the VTAM application program concepts and facilities described in this chapter apply for all types of terminals supported by VTAM. Some of the information, however, applies for only logical units and BSC and local 3270 terminals used in record mode. Users of local 3270, BSC, and start-stop terminals in basic mode should read this chapter and then read Chapter 8 for additional concerns for these terminals.

### The VTAM Language

VTAM's services are obtained by using VTAM macro instructions. VTAM macro instructions request that VTAM:

- Establish and terminate connection between the application program and VTAM
- Establish and terminate connection between the application program and terminals
- Transfer messages and responses between the application program and terminals
- Create, modify, and interrogate control blocks that hold information that is passed between the application program and VTAM
- Provide support for connection and communication activities

The operands specified in VTAM macro instructions are in keyword rather than positional format. Some of the operands must be specified; most operands are optional. Many of the VTAM macro instructions refer to a request parameter list (RPL) that describes a request to VTAM. Operands in the macro instruction can be used to modify the RPL to describe further or alter the request.

#### *VTAM Macro Instructions*

The macro instructions are grouped into the following categories:

- Connection macro instructions
- Communication macro instructions
- Network control macro instructions (OS/VS and DOS/VS Release 33 only)
- Control-block macro instructions
- Support macro instructions

#### **Connection Macro Instructions**

These macro instructions tell VTAM that a particular application program is in operation and, subsequently, request VTAM to connect the application to one or more terminals prior to transferring data. Connection macro instructions also request VTAM to disconnect the program from one or more terminals and to disconnect the program from the VTAM system.

**OPEN:** Identifies an application program to VTAM.

**CLOSE:** Indicates to VTAM that an application program is terminating its connection with VTAM.

**OPNDST:** Requests that VTAM connect the application program to a designated terminal or list of terminals. Connection is required before communication macro instructions can be used to request data transfer with a terminal.

**CLSDST:** Requests that VTAM terminate the connection between the application program and a designated terminal.

**SIMLOGON:** Allows the application program to acquire a terminal by initiating a logon on behalf of the terminal.

### Communication Macro Instructions

VTAM provides two types of communication macro instructions: basic mode and record mode. In general, *basic mode* macro instructions are used to communicate with local 3270, BSC, and start-stop terminals; *record mode* macro instructions are used to communicate with SNA terminals. However, BSC and local 3270 terminals can be used with either mode. This section describes the record mode communication macro instructions. Basic mode communication macro instructions are described in Chapter 8.

**RECEIVE:** Requests that VTAM transfer an available message or response from a specific terminal or any one of a group of terminals to the application program's data area (if the input is data) or to one or more fields of the RPL (if the input is indicators or response information).

**SEND:** Requests that VTAM transfer a message or a response to a specific terminal. Data is transferred from an area in the application program. Indicators or response information are specified symbolically in the SEND macro instruction or in the RPL, and no output area is required.

**SESSIONC:** Requests that VTAM send to a terminal a SESSIONC command that either (1) starts or stops the possibility of exchanging messages with the SEND and RECEIVE macro instructions or (2) assists in synchronizing the message sequence numbers being kept by a terminal.

**RESETSR:** Changes the mode of receiving input from a particular logical unit or 3270 terminal. The modes are *continue-any* mode (have input from the terminal satisfy an outstanding RECEIVE that specifies input from any terminal) and *continue-specific* mode (have input satisfy a RECEIVE that specifies only that particular terminal). RESETSR also cancels outstanding RECEIVES that request input from a specific terminal.

### Network Control Macro Instructions

These macro instructions allow an authorized VTAM application program to issue VTAM network operator commands (except START and HALT) and the REPLY command and to receive network operator messages from VTAM. See "Network Operator Control from an Authorized Application Program" in Chapter 4 for a description of how a VTAM application program can control the VTAM network.

**Note:** *These macro instructions are available in OS/VS and DOS/VS Release 33 only.*

**RCVCMD:** Receives network operator messages from VTAM.

**SENDCMD:** Sends VTAM network operator commands (except START and HALT) and the REPLY command to VTAM.

## **Control-Block Macro Instructions**

These macro instructions build and manipulate control blocks required by VTAM application programs. The first section describes the control blocks and the macro instructions used to assemble them. The second section describes the macro instructions that generate and modify the control blocks during program execution.

**Declarative Macro Instructions:** These macro instructions create control blocks in the application program during assembly:

**ACB:** Creates an access method control block (ACB), which provides VTAM information about the application program in its entirety. Primarily, it names the application program and the list of exit routines associated with the program.

**EXLST:** Creates an exit list (EXLST), which contains the addresses of user-written exit routines that VTAM schedules when certain conditions occur (for example, when a logon is received from a logical unit).

**NIB:** Creates a node initialization block (NIB), which provides VTAM general information about the communication that will occur between the application program and a particular terminal. This information is provided to VTAM as part of a connection request; it remains in effect for the duration of a connection.

**RPL:** Creates a request parameter list (RPL), which describe a connection, data transfer, or related request to VTAM. On completion of the requested action, the RPL contains information that VTAM passes to the application program.

**The Manipulative Macro Instructions:** These macro instructions build or modify control blocks during program execution. If these macro instructions are used to build and modify control blocks, application programs may not need to be rewritten or reassembled to reflect changes to control blocks in future releases of VTAM. The manipulative macro instructions are:

**GENCB:** Builds an ACB, EXLST, NIB, or RPL during program execution and can initialize designated fields with specified values.

**SHOWCB:** Obtains the value or values from one or more fields of a control block and places them in an area in the application program where they can be examined. In addition to fields that are set by the application program's use of macro instruction keyword operands, a number of control block fields can be shown that are set by VTAM but that cannot be directly modified by the application program.

**TESTCB:** Tests the contents of a field against a value and sets the condition code in the program status word (PSW).

**MODCB:** Changes the contents of one or more fields by inserting specified values in the fields.

There are several different forms of the manipulative macro instructions. In addition to the standard form, there is a list form, a remote list form, a generate form, and an execute form. The nonstandard forms can be used for programs that must be reenterable or that share the parameter lists that are assembled when the macro instructions are expanded.

Instead of using the manipulative macro instructions, a VTAM application can manipulate values in control blocks by using DSECT and other assembler language instructions.

## Support Macro Instructions

VTAM provides these additional macro instructions to support connection and communication activities:

**CHECK:** Checks and, if necessary, awaits completion of a previously requested RPL-based operation, marks as inactive the RPL associated with the request (thus freeing it for further use), and, if a logical or other error is detected and a LERAD or SYNAD exit routine exists, causes the appropriate routine to be executed.

**EXECRPL:** Requests that VTAM perform an operation that is currently specified in a designated RPL. EXECRPL can be used:

To reissue a specified request. When a SYNAD exit routine is entered with a return code indicating a retrievable condition, EXECRPL can be issued with assurance that the RPL contains the desired values. Use of EXECRPL to reissue a request requires that any RPL fields that may have been changed as a result of the original request be reset.

Instead of using other RPL-based macro instructions, such as OPNDST, SEND, and RECEIVE. Prior to issuing an EXECRPL, the operation to be performed must be set in the RPL; this requires the use of the IBM-supplied RPL DSECT. Other parameters may either be set in the RPL or specified with keyword operands when the EXECRPL macro is issued. The use of EXECRPL can result in fewer instructions being executed or in less storage being taken up by macro expansions, or both. Note that EXECRPL cannot be used to issue a CHECK request or reissue a CHECK request that has failed.

**INQUIRE:** Obtains certain information that the application program may need and places it in a specified area of the program. The information that can be requested using INQUIRE includes: the logon data associated with a logon; set of session parameters associated with the request; the number of terminals currently connected to, or queued for logon to, the application program; the physical address of a 3270 screen (for use in a copy operation), and whether another application program is active or inactive and whether it is accepting logons.

**INTRPRET:** Provides access to a user-defined interpret table. For example, INTRPRET can be used to obtain the real symbolic name of an application program when the program is identified with an alias in a logon. INTRPRET can be used by programs written to receive logons and reconnect terminals to the appropriate application program.

**SETLOGON:** Tells VTAM to activate the application program's LOGON exit routine.

**SIMLOGON:** Allows the application program to acquire a terminal by initiating a logon on behalf of the terminal.

## *Relating VTAM Control Blocks and Executable Macro Instructions*

The control blocks created by VTAM macro instructions are used to pass information between the application program and VTAM. One control block, EXLST, is used simply to pass the names of exit routines in the application program to VTAM. The other control blocks, the ACB, RPL, and NIB, are used both to pass information to VTAM and to contain information that VTAM returns to the program.

The most frequently used control block is the RPL. An RPL is required each time the program makes a request for a connection, for an I/O operation, or for any service except opening and closing an ACB. The other control blocks, the ACB, EXLST, and NIB, may be used as infrequently as once during the execution of the program.

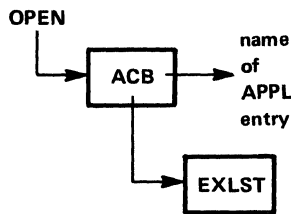
Each RPL-based macro instruction refers to an RPL that contains information about the operation to be performed, such as the address of the input or output area, the length of

the area, and the terminal to be communicated with. This information is placed in the RPL initially when the control block is created or subsequently, either by using a MODCB macro instruction or by providing the information in the macro instruction itself. For example:

```
RECEIVE RPL=RPL1,AREA=INFO,AREALEN=200
```

changes the value in RPL1's AREA and AREALEN fields to INFO and 200 respectively before initiating the input operation. These values remain in these fields for all subsequent requests that used RPL1 until they are changed again.

### Opening and Closing the Application Program



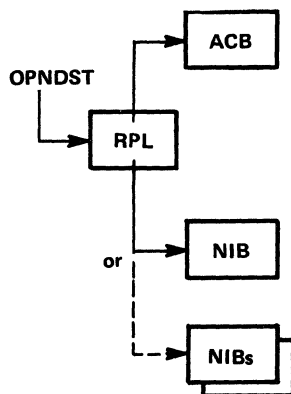
The OPEN macro instruction indicates to VTAM that a program is an operational part of the VTAM system. The OPEN specifies an ACB; the ACB in turn points to a location in the program that contains the name of the application program as defined in an APPL statement during VTAM definition. The ACB may also point to an EXLST control block containing the names of exit routines that are to be associated with the application program. When the open process is completed, any exit routines that have been specified are eligible for scheduling by VTAM.

A single OPEN macro instruction can open more than one ACB at a time. This means that a program that performs related functions (for example, communicating with both logical units and other terminals) may be defined so that it is viewed by VTAM as more than one application program. Many VTAM users will find it satisfactory to open only one ACB for each program.

The CLOSE macro instruction notifies VTAM that an application program is detaching itself from the VTAM system. A single CLOSE macro instruction can close more than one ACB. The CLOSE macro instruction disconnects any terminals still connected to the program.

### Connecting Terminals

Before communicating with a terminal, an application program must have itself connected to the terminal. Connection can be initiated by the terminal, the network operator, VTAM, another application program, or the application program. In any case, the connection is not effective until the application program issues an OPNDST macro instruction. The OPNDST macro instruction specifies an RPL that is associated with the request. The RPL contains the address of a NIB. The NIB contains information that applies to subsequent communication with the terminal; this information can include a unique storage address to be associated with the terminal. If a number of terminals are to be connected, a single OPNDST can be used with the RPL pointing to a list of NIBs instead of to a single NIB.



Optionally, a NIB can point to a list of exit routine names in an EXLST control block. For the terminal being connected, these exit routines are used in preference to any equivalent exit routines identified when the ACB was opened.

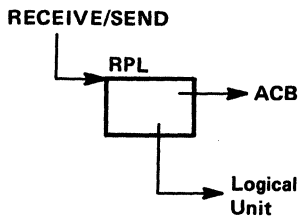
When a terminal is connected as the result of an OPNDST macro instruction, VTAM returns information about the terminal in the RPL and the NIB. In both the RPL (if only one terminal is connected) and the NIB, VTAM places a communication identifier (CID) that it has assigned to the terminal. It is shorter and requires less space and search time for VTAM than the symbolic name of the terminal specified during VTAM definition. On all subsequent I/O requests for the terminal, the application program must be sure that this CID is located in the RPL. In addition to the CID, VTAM also places the terminal name, characteristics, and other information in the NIB; if desired, the application program can use this information to determine how to communicate with the terminal.

Once a NIB has been used to connect one terminal, it can be reinitialized and reused for connection with other terminals.

### Disconnecting Terminals

The program uses the CLSDST macro instruction to disconnect a terminal. If the program is terminating and all terminals are to be disconnected at the same time, the CLOSE macro instruction, used to close the ACB, also disconnects all connected terminals, and the CLSDST macro instruction need not be used.

### Communicating with Terminals in Record Mode



Having opened the ACB and connected one or more logical units or BSC or local 3270 terminals, the program can communicate with each connected terminal by issuing SEND and RECEIVE macro instructions. (Communicating with terminals in basic mode is described in Chapter 8.) VTAM obtains the name of the application program that made the request and the identity of the terminal (if a specific terminal is being addressed) from the RPL. The data transfer macro instruction specifies an RPL; the RPL contains the address of an ACB and the terminal's CID.

Only data is written from or read into an application program data area. Indicators and response information are sent as a result of being specified symbolically in a SEND macro instruction or in an RPL. Indicators and response information that are received are not read into a data area but are detected by analyzing fields in the RPL associated with a RECEIVE macro instruction or in an RPL associated with the scheduling of an exit routine specified to handle the receipt of indicators or response information.

VTAM provides two ways to send to a terminal in record mode. A SEND macro instruction can be issued that specifies POST=RESP, in which case VTAM handles any response signals that are returned and notifies the application program when the requested operation is completed; this is called *responded output*. (If necessary, the application program can obtain response information from the RPL.) A SEND macro instruction can also be issued that specifies POST=SCHED, in which case VTAM considers the SEND complete as soon as the output operation has been scheduled; this is called *scheduled output*. The application program thus has the responsibility for determining completion either by having a RECEIVE macro instruction that specifies response information is to be received or by having a RESP exit routine that VTAM schedules each time response information arrives.

### Handling Control Blocks, I/O Areas, and Work Areas

The application program can handle control blocks in a number of ways. It can:

- Define RPLs, NIBs, OR EXLSTs in the application program during assembly or generate them, using the GENCB macro instruction, during program execution
- Assign one RPL or NIB to a specific terminal during assembly, or either assemble or generate RPLs and NIBs that are to be available for any terminal as the need arises
- Retain the RPL used in connecting the terminal for all further communication with the terminal
- Use one RPL for all connection requests and use another RPL or group of RPLs for all data transfer requests
- Define the RPLs, NIBs, and any other required control blocks or work areas to be associated with terminals as a pool, so that a limited amount of control block storage is not exceeded

In application programs that must handle many terminals concurrently, it may be useful to have a control block work area other than the RPL or NIB associated with a particular terminal or terminal session. The queuing of input or output by the application program between its processing routines and terminals may also require a terminal-related control

block work area. The RPL contains a USER field that may be used to associate storage with a particular terminal. The application program initially associates the storage with the terminal when the terminal is connected by specifying an address in the USERFLD of the NIB; thereafter, when input is received from the terminal, VTAM provides the address associated with it in the RPL's USER field.

## Organization of the Application Program

VTAM provides facilities that allow an application program to be organized in a variety of ways. This section describes some of the major VTAM facilities that affect program organization. See the *VTAM Macro Language Guide* for a description of other considerations.

### *Overlapping VTAM Requests with Other Processing*

Each request issued by the application program can be handled synchronously or asynchronously by VTAM.

*Synchronous* request handling means that VTAM returns control to the next sequential instruction only after the requested operation has been completed. Program execution is halted until VTAM determines that the operation has been completed. This type of request handling is appropriate for application programs that cannot continue processing until a particular request has been completed. Figure 5-1 illustrates synchronous request handling.

*Asynchronous* request handling means that VTAM returns control to the next sequential instruction as soon as VTAM has accepted the request, not when the requested operation has been completed. Accepting a request consists of screening the request for errors and scheduling the parts of VTAM that will eventually carry out the requested operation. While the operation is being completed, the application program is free to initiate other I/O transactions or processing. For example, an application program might issue a RECEIVE macro instruction and indicate that it is to be handled asynchronously; while the input operation is being completed, the application program can begin to write to a direct access storage device or communicate with another terminal.

When asynchronous request handling is used, there are two ways that VTAM can notify the application program that the requested operation has been completed. If the application program associates an event control block (ECB) with the request, VTAM posts the ECB when the operation is completed. The application program can use a CHECK or WAIT macro instruction to determine whether the ECB has been posted. Alternatively, the application program can associate an RPL exit routine with the request.

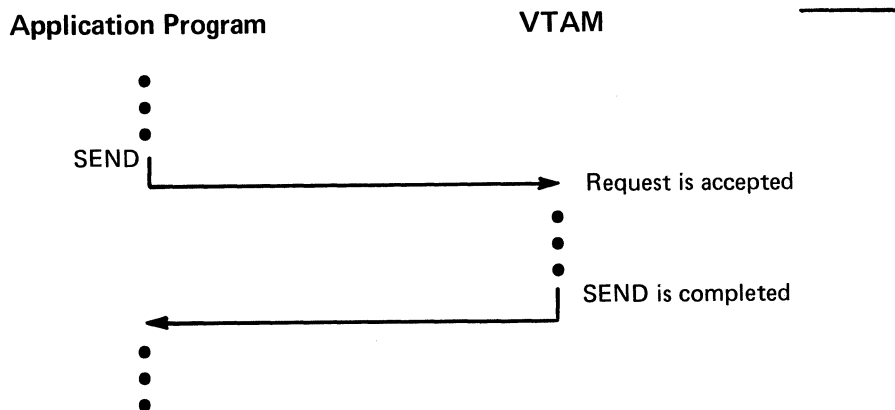


Figure 5-1. Processing Pattern for a Synchronous Request



When the operation is completed, VTAM invokes the routine. Figure 5-2 illustrates asynchronous processing in an application program using ECBs; Figure 5-3 illustrates the use of the RPL exit routine to control processing.

By using ECBs, the application program can use one WAIT macro instruction for a combination of VTAM requests and any non-VTAM requests that use ECBs. For example, an application program can issue three VSAM requests and three VTAM requests; by issuing one WAIT for all six ECBs, the application program can continue processing when any one of the six operations is completed.

ECBs also give the application program the freedom to determine during program execution when the program should stop processing and wait for a given operation to be completed. An application program might, for example, request data from a terminal and then later determine that it should not stop and wait for that data (perhaps the application program has in the meantime begun to request data from another terminal whose input is of a much higher priority and must be handled immediately). ECBs also allow the application program to assign priorities to requests by checking some ECBs before checking others.

The difference between using ECBs and RPL exit routines is that an RPL exit routine is *automatically* scheduled when the requested operation is completed, thereby saving the application program the trouble of checking ECBs and branching to subroutines. ECBs provide greater control, while RPL exit routines are more convenient.

VTAM requests issued in the RPL exit routine can also be handled asynchronously, although an exit routine is not scheduled if another exit routine has not completed execution. Figure 5-4 illustrates how an application program might use this facility.

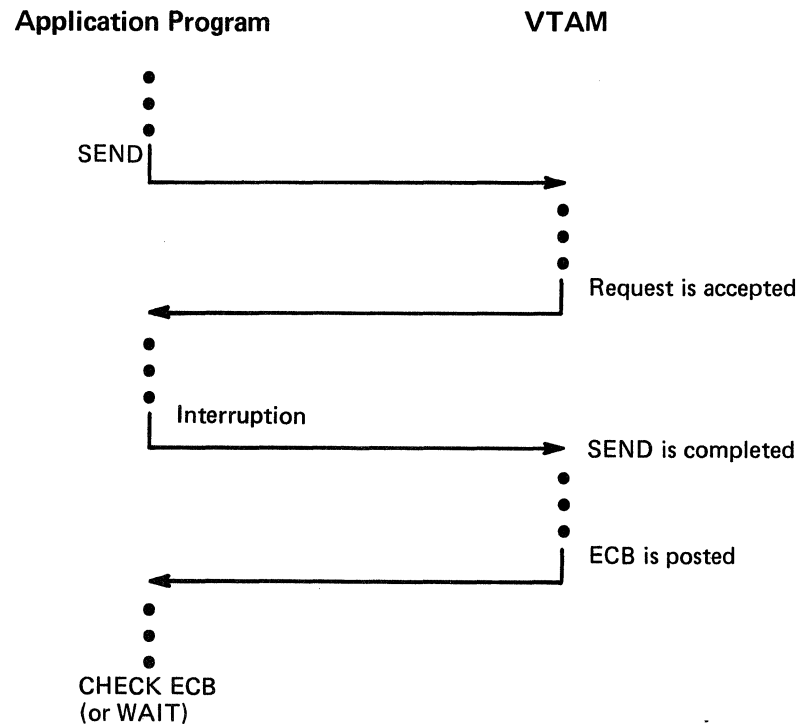


Figure 5-2. Processing Pattern When an ECB Is Used with an Asynchronous Request

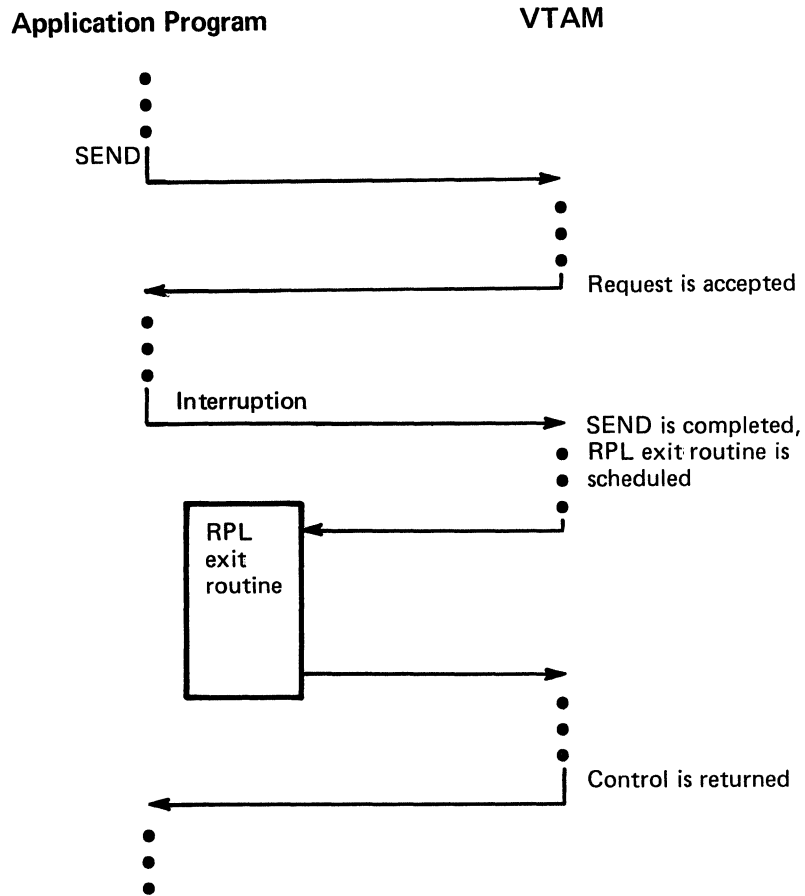


Figure 5-3. Processing Pattern When an RPL Exit Routine Is Used with an Asynchronous Request

### *EXLST Exit Routines*

The application program can use the EXLST macro instruction to identify a variety of exit routines that VTAM schedules when particular events occur:

Event	Routine
A terminal is waiting to be connected to the application program	LOGON
Special notification regarding connection status	LOSTERM
A connected terminal is wanted by another application program	RELREQ
VTAM is being shut down	TPEND
The application program has made an invalid connection or I/O request	LERAD
A physical error has occurred during a connection or I/O operation	SYNAD
A start-stop terminal has caused an attention interruption	ATTN
A special type of input has arrived from a logical unit (the types of input are discussed later)	DFASY RESP SCIP

Note that DFASY, RESP, and SCIP exit routines are not used with local 3270, BSC, and start-stop terminals.

## Application Program

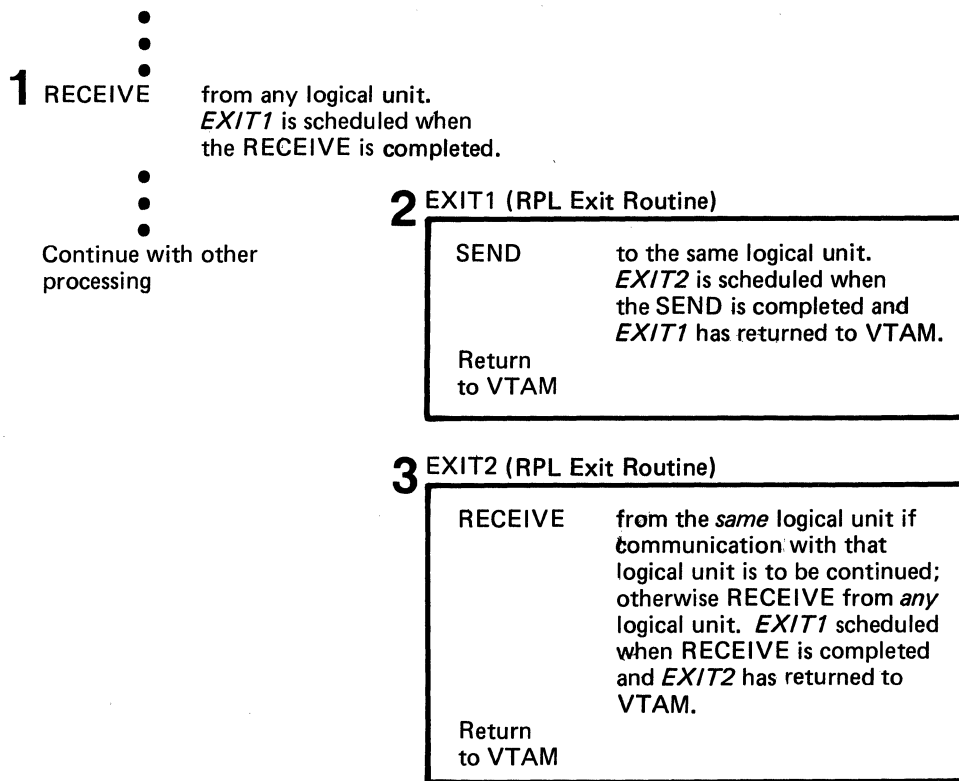


Figure 5-4. A Possible Processing Pattern When Asynchronous Requests Are Issued in RPL Exit Routines

When one of these events occurs, the execution of the application program is interrupted and the appropriate exit routine is given control. If another event occurs while the exit routine is being executed, the next exit routine is not invoked until the first has completed execution (this applies as well for RPL exit routines).

Unlike accounting, authorization, and logon-interpret exit routines (discussed in Chapter 3), which are included as part of the system during VTAM definition, the EXLST exit routines are included as part of the application program. The addresses of these routines are placed in an EXLST control block by the application program.

In OS/VS, each exit routine is usually scheduled under an interruption request block (IRB); in DOS/VS, each exit routine is scheduled by changing the program information block (PIB) save area address. Any processing in the routine that places the routine in a wait state should be used with caution, since the application program's entire task waits while the exit routine waits. Exit routines other than the LERAD and SYNAD exit routines need be reenterable only if two or more application programs share the same exit routine.

### **Error Notification**

When synchronous request handling is used, error conditions are reported when the request has been completed and control returned to the application program. When asynchronous request handling is used, error conditions are reported in two stages. When control is first returned to the application program, VTAM indicates whether it has accepted or rejected the request. When an accepted request has been completed, VTAM posts an ECB or schedules a designated RPL exit routine and, on the program's issuing a

CHECK macro instruction, returns information about the completion of the requested operation. Figure 5-5 shows how errors are reported during asynchronous processing.

Information about success or failure is sent to the VTAM application program in register 15 and, under some circumstances, in register 0. If the request is rejected or the operation is successful, VTAM normally places additional information in return code fields of the RPL and attempts to enter one of the two types of error-handling exit routines that the user furnishes. The application program can code two error-handling routines that VTAM attempts to invoke as a result of all physical, environmental, and logical errors. The routine that handles logical errors is called the LERAD routine, and the routine that handles physical errors detected by VTAM is called the SYNAD routine.

Should an error occur during an RPL-based operation that is specified as synchronous, the LERAD or SYNAD routine is invoked as part of the processing. Should an error occur in conjunction with an operation that is specified as asynchronous, the LERAD or SYNAD exit routine is entered at either of two times: as a result of the request being rejected or, if the request is accepted, as the result of a CHECK being issued. When the LERAD or SYNAD routine receives control, it is provided in register 0 and in the RTNCD field of the RPL with information regarding the specific cause of the error.

The VTAM application program, in the main program or in a LERAD or SYNAD exit routine, can associate a class of completion information with a particular action. VTAM organizes its setting of register 0 and the RTNCD field of the RPL into these completion categories:

**Extraordinary completion:** This requires further analysis of the RPL to determine the course of action required.

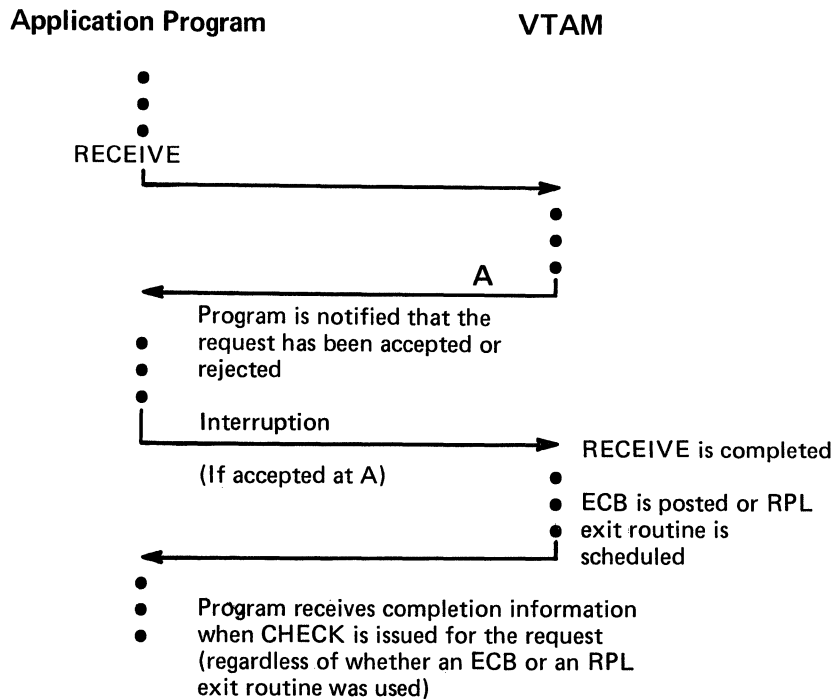


Figure 5-5. Processing Pattern for Reporting Errors during an Asynchronous Operation

**Retriable completion:** This indicates the request can be reissued. The EXECRPL macro instruction can be used.

**Damage completion:** This indicates that, in addition to reissuing a request, some input or output data may have to be recovered or corrected.

**Environment error completion:** This indicates that the program should call for external intervention.

**User logic error completion:** This indicates that the program status should be saved, perhaps by requesting a dump. Depending on the seriousness of the error, the program can continue or terminate.

In some cases, determining that the return code (in register 0 and the RTNCD field of the RPL) specifies one of these completion categories can determine the course of action. In other cases, additional analysis of the RPL and other information (such as program flags) is required to determine what action the program should take.

**LERAD Processing:** Since most logical errors result from flaws in the design of an application program, the handling of logical errors is most important during the development and debugging of the application program. The handling of the error may consist of little more than recording the attributes of the request that failed, requesting a dump, and causing an abnormal termination.

**SYNAD Processing:** Physical and environmental errors usually require more extensive treatment. The application program should, during program execution, determine the nature of the error, assess the extent of the problem, and attempt remedial action. If the application program determines that the error occurred because incoming data exceeded the capacity of the NCP buffers, for example, the application program could inform the logical unit that the data should be resent in two transmissions. If the problem is a recurring hardware check for the logical unit, however, the application program may have to take whatever action is required to continue without the logical unit.

The error-handling routine can set register 0 or register 15 and return through VTAM to the instruction following the synchronous request or CHECK macro instruction. If the exit routine corrects the error, register 15 and, in some cases, register 0 can be set to 0 to indicate that the requested operation completed normally. The main part of the program continues normally, unaware that an error or special condition occurred and that the LERAD or SYNAD exit routine was entered. If, however, the exit routine is not able to correct the error, it sets a value in either or both of registers 15 and 0 before returning, so that the main program can take any action that the LERAD or SYNAD exit routine did not take.

**Note:** *The LERAD and SYNAD routines must be coded to be reenterable unless all RPL-based requests are in the main program or all RPL-based requests are in asynchronous exit routines. Unlike other exit routines, the LERAD and SYNAD exit routines can be interrupted and reentered as the result of RPL-based requests within themselves or in other parts of the program.*

### ***Other Considerations for Program Organization***

In addition to the facilities described above, a VTAM application program can use VTAM and operating system facilities that allow multithreading, multitasking, and use of multiple ACBs. In OS/VS2 MVS, a VTAM application program can specify execution of individual SEND, RECEIVE, and RESETSR macro instructions in a path that requires fewer instructions (authorized path). See the *VTAM Macro Language Guide* for a description of these facilities.

## Application Program Concepts and Facilities

VTAM's primary purpose is to provide communication between the application program and terminals. This section discusses the application program's ability to have a terminal allocated to it and its ability to communicate with that terminal.

### *Opening the Application Program*

Before using any VTAM facilities, a VTAM application program must issue an OPEN macro instruction to identify itself to VTAM. A VTAM application program typically begins with an OPEN macro instruction and ends with a CLOSE macro instruction. That is, the association between the application program and VTAM normally lasts for the duration of the application program's execution.

The application program also uses the OPEN macro instruction to supply VTAM with the addresses of its error-handling routines (LERAD and SYNAD) and its event-driven exit routines (such as LOGON, LOSTERM, and RELREQ).

### *Connection*

An application program must issue an OPNDST (open destination) macro instruction to establish connection with a terminal before communication with that terminal can begin. The OPNDST macro instruction causes VTAM to "allocate" the terminal to the application program. OPNDST initializes VTAM's and the application program's control blocks to indicate that the terminal and the application are connected. OPNDST also ensures that an active path exists between the two nodes before connecting them. Unlike the effect of an OPEN macro instruction, the effect of an OPNDST often does not last for the duration of the application program's execution. Because of VTAM's terminal-sharing facility, connections to the network's terminals can be made, broken, and remade throughout the duration of the application program's execution. An application program can establish connection either by *accepting* the terminal or by *acquiring* the terminal.

### *Acceptance*

When an application program accepts the terminal, it does so because a logon was received for the terminal. The acceptance does not complete unless (or until) a logon has been issued for it. Although there are several possible sources of the logon—the network operator, another application program, the terminal itself, or VTAM—the request usually originates *outside* of the application program. (Logons requests can also be generated *within* the application program; however such logons are essentially a form of acquisition and are discussed in "Acquisition" later in this chapter.)

Acceptance is suitable for application programs that do not require access to a specific terminal or specific set of terminals in order to function, but instead are designed to service various terminals that require access to the application program. If the user wants the terminals themselves to designate which application program they are to use, the user can allow each terminal to initiate logons so that the application program can accept the terminal.

**Note:** *When VTAM notifies an application program of a logon, the terminal and its logon are queued to the application program. As long as the terminal is queued to the application program, it is not available for connection to other programs; it is available for connection only to the application program to which it is queued. The application program and its queued terminal are unable to communicate with each other until the connection is completed by the program's acceptance of the terminal. Because a queued terminal is effectively eliminated from the system until accepted or disconnected by the application, the user should ensure that application programs avoid leaving terminals on this queue any longer than necessary.*

**Accepting Terminals with an Exit Routine:** The application program can maintain a LOGON exit routine that VTAM schedules whenever a logon for the application program is received. VTAM provides the exit routine with the identity of the terminal associated with the logon. The application program can either accept the terminal (using an OPNDST macro instruction) or reject it (using a CLSDST macro instruction).

The application program does not have to use an exit routine in order to determine when a logon has occurred. The application program can instead issue a connection request that VTAM completes as soon as a logon occurs. Although this method is simpler to use than an exit routine, the application program does not have the opportunity to decline the logon prior to establishing connection with the terminal.

**Preventing Logons:** Before an application program issues a SETLOGON macro instruction, VTAM will queue logons but will not schedule the application program's LOGON exit routine. After the SETLOGON is issued, the LOGON exit routine will be scheduled for each logon. An application program can issue another SETLOGON macro instruction to stop the scheduling of the LOGON exit routine or the queuing of logons.

**Types of Acceptance:** The application program can issue a connection request to accept a *specific* terminal, or to accept *any* terminal for which a logon has been issued. To accept a specific terminal, the application program must tell VTAM the identity of the terminal; connection is not made until a logon has been issued for that particular terminal. The application program can also accept a logon from any terminal in the network. After connection is established, VTAM passes the identity of the terminal to the application program.

## Acquisition

An application program acquires a terminal or set of terminals by issuing an OPNDST macro instruction with the ACQUIRE option or a SIMLOGON macro instruction. OPNDST with the ACQUIRE option connects the terminal in one step; SIMLOGON generates a logon on behalf of a terminal and the application program must accept the terminal. The user must authorize the application program's use of acquisition when defining the application program to VTAM.

An application program acquires one or more terminals from a set by building a series of NIBs, each NIB containing the name of a terminal, and then indicating which terminals from the set are to be acquired. The CONALL option specifies that as many terminals from the set as are available are to be acquired. The CONALL option is used when the application program is willing to proceed with as many active terminals as are available (that is, not already connected to another application program). VTAM provides information so that the program can determine which terminals were available.

An application program requests any *one* terminal from a set using the CONANY option. The first available terminal from the set is acquired. This type of acquisition is useful for application programs that require a terminal, but for which any terminal is as good as another.

One application program cannot take another application program's terminals from it. If an application program attempts to acquire a terminal that is already connected to another application program, no reconnection is possible until the current owner disconnects it. VTAM allows an application program to request another application program's terminal by issuing a SIMLOGON macro instruction that indicates that the current owner is to be notified of the request. (OPNDST with the ACQUIRE option cannot request that the current owner be notified of the request.) The application program should request this notification if it needs the terminal regardless of its

connection status. Notification should not be indicated when the requesting application program only needs the terminal if it is unconnected.

The owning application program also controls whether it can be notified when another application program issues a connection request to acquire one of its terminals. Notification can therefore only occur when both application programs so indicate. VTAM notifies the owning application program by scheduling its RELREQ exit routine.

The RELREQ exit routine is provided with the identity of the contested terminal. The application program can elect to disconnect the terminal immediately, disconnect it later, or ignore the request entirely. If the terminal is disconnected, the previous owner can immediately attempt to re-acquire the terminal from the new owner (using a queued connection request as described below) so that it will be returned when it is no longer being used. When the terminal is disconnected from the new owner, it is reconnected to to the acquiring application program that has waited the longest, not necessarily to the application program that was the previous owner.

By controlling which application programs release contested terminals and which do not, the user can cause some application programs to be able to obtain and keep terminals more readily than other application programs. Or, the user can establish a policy that all application programs release sharable terminals that are not being used.

### **Establishing Sets of Session Parameters**

When an application program establishes connection to a terminal that uses record mode, it can specify a set of *session parameters* (rules that an application program and a terminal agree to follow when communicating). Local and BSC 3270 terminals used in record mode use a set of session parameters supplied by VTAM; any session parameters specified during connection are ignored. Local 3270, BSC, and start-stop terminals used in basic mode do not use session parameters; any session parameters specified during connection are ignored. Some of the parameters specified by these session parameters are: message/response conventions, acceptable types of chaining, error-recovery responsibility, and bracket usage. Some of these parameters are discussed later in this chapter. *VTAM Macro Language Guide* describes session parameters used by a VTAM application program. To specify a set of session parameters, an application program can:

- Specify a predefined set of session parameters by supplying a logon mode name in the NIB associated with the OPNDST macro instruction. VTAM uses this name to search a logon mode table for the desired session parameters.
- Specify that the session parameters contained in its storage area (pointed to by the NIB) are to be used. IBM supplies a DSECT for formatting the storage area.
- Indicate that the set of session parameters associated with the pending logon is to be used (OPNDST with OPTCD=ACCEPT only). Included in a logon (whether caused by a terminal action, network-operator action, a SIMLOGON macro instruction, or a CLSDST macro instruction) is a logon mode name. For logical units, VTAM uses the logon mode name to search a logon mode table for the associated session parameters to be used if the application program requests them (local and BSC 3270 terminals used in record mode use a fixed set of session parameters). An application program can examine the session parameters associated with a pending logon by using the INQUIRE macro instruction.

### **Queuing Connection Requests**

Application program requests for connection always fail if the terminal is inactive. If the terminal is active but unavailable, however, the application program designates whether its connection request should fail or whether the request should remain pending (queued) until the terminal does become available.



Although the definition of an “available” terminal differs between acquired and accepted connections, the option of queuing or not queuing the communication request applies to both. When an application program attempts to *acquire* an active terminal, the terminal is available if it is not connected (or queued for connection as the result of a logon) to another application program. When an application program attempts to *accept* a terminal, the terminal is available if a logon for it has been directed at the application program. (Note the distinction between queuing an application program’s request for connection, described here, and queuing a terminal to an application program as the result of a logon, as described in “Acceptance” earlier in this chapter.)

Figure 5-6 lists the effects of queuing a connection request.

### ***Disconnection***

To disconnect a terminal, an application program can *release* it or it can *pass* it to another application program. Passing and releasing are accomplished by using the PASS and RELEASE options of a CLSDST macro instruction. The terminal is released by disconnecting it without regard to which application program (if any) is to receive it. The terminal is passed by disconnecting it and designating which application program is to receive it. Passing must be authorized when the application program is defined to VTAM.

If the terminal is released, VTAM connects it to any application program that has issued a SIMLOGON macro instruction for it and indicated that its connection request should be queued. If more than one application program has attempted to acquire the terminal, VTAM connects it to the application program that first issued the connection request. If there are no queued SIMLOGONS, VTAM generates an automatic logon for the terminal if automatic logon was specified for the terminal during VTAM definition. If automatic logon was not specified, the terminal remains unconnected.

When a terminal is passed, VTAM disconnects it, generates a logon and directs the logon to the designated application program. The terminal is not reconnected until the receiving application program accepts it. A terminal should be passed only when it is imperative that it be connected to a specific application program. For example, a user might maintain several application programs that each require the same information from the terminal before any of the programs can continue execution. Although each application program could conduct its own interrogation, it might be simpler to maintain a single application program that converses with the terminal to obtain the initial information and then passes the terminal to the appropriate destination.

When the application program passes a terminal, it can also pass data to the receiving application program. In the example above, the application program might pass along the results of the preliminary conversation.

### ***Record-Mode Communication***

VTAM provides two modes of data transfer, basic and record. Basic mode is always used for start-stop and BSC (except 3270) terminals. Record mode is used for logical units. Local and BSC 3270 terminals can be used in either mode. This section describes record-mode communication. The section “Basic-Mode and Record-Mode Communication” describes communication concepts that are common to basic and record mode. Chapter 8 describes basic-mode communication in detail.

Communicating with terminals in record mode requires an understanding of concepts related to:

What is communicated between a VTAM application program and a terminal (messages and responses)

Type of Connection Request	Meaning When Connection is to be Queued	Meaning When Connection is not to be Queued
<p>OPNDST ACCEPT: A specific terminal (SPEC)</p> <p>Any terminal (ANY)</p>	<p>Connect the specified terminal if a logon request has been received for it. Otherwise, connect the terminal when a logon is received for it.</p> <p>Connect any terminal for which a logon has been received (if logons have been received for more than one terminal, connect the terminal that has waited the longest). Otherwise, wait until a logon is received from any terminal and then connect that terminal.</p>	<p>Connect the specified terminal if a logon has been received for it. Otherwise, indicate failure in the return code.</p> <p>Connect any terminal for which a logon has been received (if logons have been received for more than one terminal, connect the terminal that has waited the longest). Otherwise, indicate failure in the return code.</p>
<p>OPNDST ACQUIRE: A set of one (CONANY)</p> <p>Any one of a set (CONANY)</p>	<p>(Cannot be queued)</p> <p>(Cannot be queued)</p>	<p>Connect the terminal if it is available. Otherwise, indicate failure in the return code.</p> <p>Connect the first terminal in the set (NIB list) that is available. Otherwise, indicate failure in the return code.</p>
<p>OPNDST ACQUIRE: As many as are available in a set (CONALL)</p>	<p>(Cannot be queued)</p>	<p>Connect all terminals in the set (NIB list) that are available. If none are available, indicate failure in the return code.</p>
<p>SIMLOGON: A set of one (CONANY)</p> <p>Any one of a set (CONANY)</p> <p>As many as are available (CONALL)</p>	<p>A. If the terminal is available and an OPNDST ACCEPT has been queued for this terminal, connect the terminal.</p> <p>B. If the terminal is available but no OPNDST ACCEPT has been queued for it, schedule the LOGON exit routine.</p> <p>C. If the terminal is not available, queue the SIMLOGON request until the terminal becomes available, and then perform A or B above.</p> <p>A. For the first terminal in the set (NIB list) that becomes available, connect the terminal if an OPNDST ACCEPT has been queued for this terminal or <i>any</i> terminal.</p> <p>B. If an OPNDST ACCEPT has not been queued for the terminal or <i>any</i> terminal, schedule the LOGON exit routine for the first available terminal in the set (NIB list).</p> <p>C. If no terminal in the set (NIB list) is available, queue a SIMLOGON request for each terminal in the set. When the first terminal becomes available, do A or B above, as appropriate, and dequeue the other SIMLOGON requests.</p> <p>A. For each available terminal in the set (NIB list), either (1) connect the terminal if an OPNDST ACCEPT has been queued for the terminal or <i>any</i> terminal, or (2) schedule the LOGON exit routine if no OPNDST ACCEPT has been queued.</p> <p>B. For each terminal in the set (NIB list) that is not available, queue a SIMLOGON request for that terminal. When the terminal becomes available, connect it if an OPNDST ACCEPT has been queued for it or for <i>any</i> terminal, or schedule the LOGON exit routine if no OPNDST ACCEPT has been issued.</p>	<p>A. If the terminal is available and an OPNDST ACCEPT has been queued for this terminal or <i>any</i> terminal, connect the terminal.</p> <p>B. If the terminal is available, but no OPNDST ACCEPT has been queued for it, schedule the LOGON exit routine.</p> <p>C. If the terminal is not available, indicate failure in the return code.</p> <p>A. For the first terminal in the set (NIB list) that becomes available, connect the terminal if an OPNDST ACCEPT has been queued for this terminal or <i>any</i> terminal.</p> <p>B. If an OPNDST ACCEPT has not been queued for the terminal or <i>any</i> terminal, schedule the LOGON exit routine for the first available terminal in the set (NIB list).</p> <p>C. If no terminal in the set (NIB list) is available, indicate failure in the return code.</p> <p>A. If all terminals in the set (NIB list) are available, either (1) connect each terminal if an OPNDST ACCEPT has been queued for it or <i>any</i> terminal, or (2) schedule the LOGON exit routine for each unit if no OPNDST ACCEPT has been queued.</p> <p>B. If any terminal in the set (NIB list) is not available, indicate failure in the return code.</p>

Figure 5-6. Queued and Unqueued Connection Requests

VTAM facilities for sending and receiving messages and responses

Protocols that may be used to control the exchange of messages and responses

## Messages and Responses

Communication between a VTAM application program and a terminal in record mode consists of an exchange of *messages* and *responses*. A *message* consists of data and control information or sometimes control information alone. For example, a message can consist of data (perhaps an answer to an inquiry previously forwarded) furnished by the VTAM application program in a designated data area, and other information, such as the kind of acknowledgment wanted, that is specified symbolically to VTAM when the program requests VTAM to send the message. Or, in some cases, a message can contain no data, and be intended to control the further exchange of data. For example, the program could specify that VTAM send a message on its behalf to a terminal indicating that the terminal should stop sending until released to send by the VTAM application program.

A *response* indicates whether or not a message arrived successfully. A response is positive (the message arrived and is acceptable) or negative (the message did not arrive or is not acceptable).

A message or a response is sent by the VTAM application program by issuing a SEND macro instruction. For a message, the SEND specifies a data area (if there is data) and control information. For a response, the SEND specifies the nature of the response (positive or negative) and possibly control information for a response. A message or a response is received by a VTAM application program by issuing a RECEIVE macro instruction. (In some cases, input can also be received as the result of VTAM's scheduling a special-purpose exit routine; in this case, the input is present when the exit routine is entered and a RECEIVE does not have to be issued.)

When VTAM receives a SEND macro instruction request, it moves the data from the VTAM application program to its own buffers and, together with the control information specified in the SEND macro instruction, formats the message.

## Message Flow: Normal and Expedited

The VTAM application program can send and receive messages in two different message streams or flows. *Normal-flow* messages are messages that contain data and certain control commands; *expedited-flow* messages are messages that contain control commands that require that they be handled ahead of the normal-flow messages. For example, if a VTAM application program wants to signal to a logical unit to shut down its operations for the day, it can send a Shutdown command message; this message will be sent ahead of other messages that are scheduled to be sent from the VTAM application program to the logical unit.

## Levels of Message Control: Session Control and Data Flow Control

Ordinarily, a VTAM application program and a terminal are in a level of message control in which data and related control information and responses are exchanged. Using special commands, however, a VTAM application program can stop and restart the exchange of data messages and related control information and responses. While this flow is stopped, the VTAM application program and the terminal, using special *session control* commands, can exchange information about *sequence numbers* with which each data message is associated. (Each normal-flow message is assigned a sequence number by VTAM; this number is one greater than the sequence number assigned the previous normal-flow message that the VTAM application program requested be sent to the terminal.)

The SESSIONC macro instruction is used to start, stop, and exchange information about sequence numbers. Figure 5-7 shows an example of SESSIONC used to stop and restart the data flow.

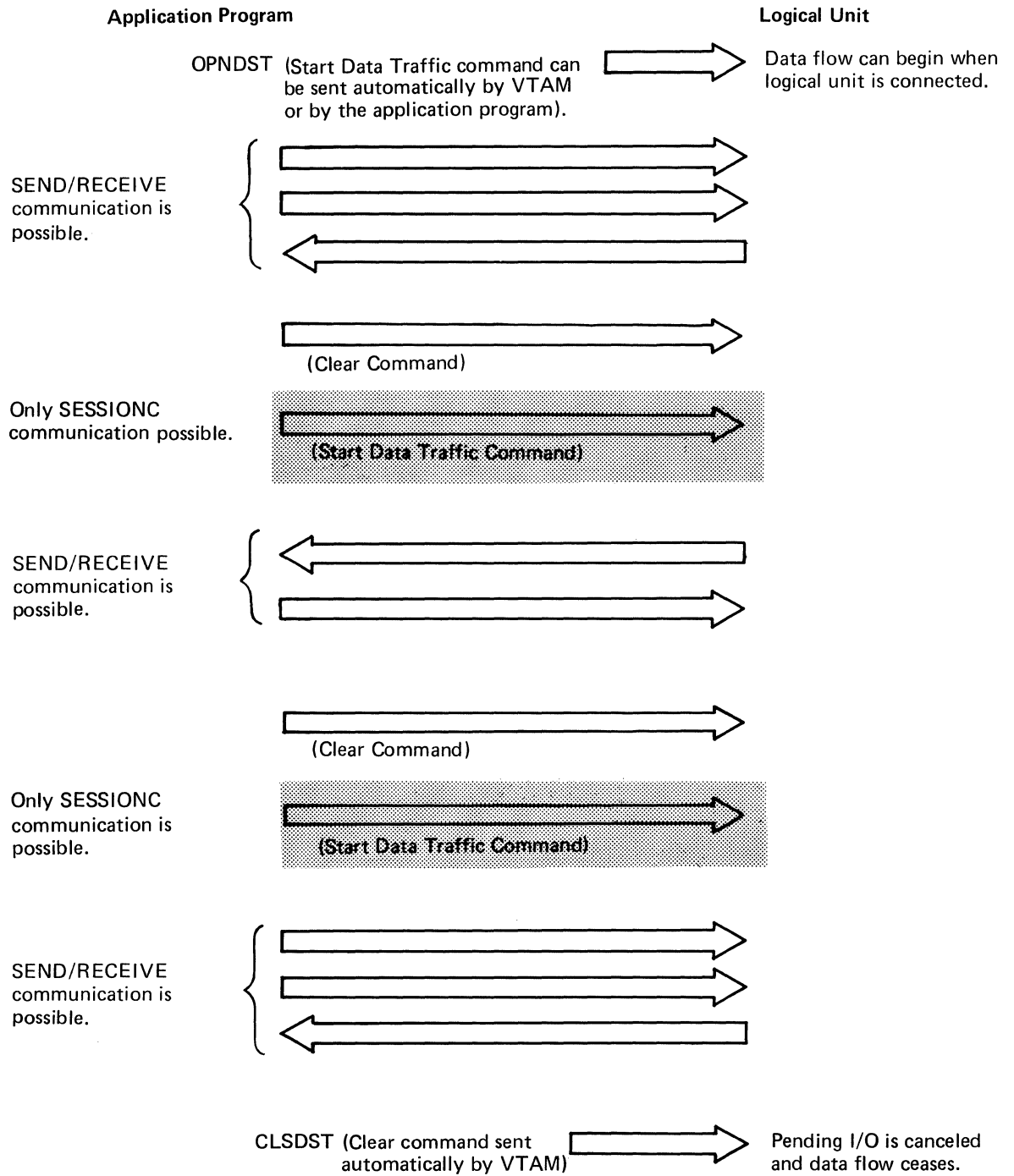


Figure 5-7. An Example of Start Data Traffic and Clear Commands

**Responses: Requesting a Response**

When the VTAM application program or a terminal sends a message, it specifies the conditions under which it wants a response to that message. The message sender can request a definite response (it wants either a positive or a negative response as appropriate) or an exception response (it wants a response only if a negative response is appropriate). The receiver of the message must respond as appropriate. The VTAM application program requests a definite response, an exception response, or no response by specifying a parameter in the RESPOND field of the RPL. The VTAM application program determines the type of response required after it has received a message by examining the RESPOND field of the RPL. Figure 5-8 shows a case where a definite response is requested and a case where an exception response is requested.

Requesting definite responses, rather than just exception responses, results in greater control over error conditions and provides an opportunity for quicker error recovery, but it requires more programming and more traffic in the network. A request for only exception responses can be used within a group of related output messages if the entire group is to be discarded when an error condition occurs. Requesting no responses of either sort is appropriate when there is no concern for error recovery for that transmission.

**Responses: Positive or Negative**

When a definite response has been returned to a message, the response receiver determines whether the response is positive or negative. If only negative responses were requested, the receipt of any response would indicate a negative response.

A VTAM application program receives a response by means of a RECEIVE macro instruction or an RESP exit routine. The type of response received through a RECEIVE macro instruction can be determined by examining the RESPOND field of the RPL associated with the RECEIVE. A response received through an RESP exit routine is identified by VTAM as positive or negative when the routine is invoked.

A terminal or an application program can send a negative response to a message even though it arrived successfully. For example, a negative response might indicate that the data in the message was not in a prescribed form.

**Responses: Definite Response 1 and 2**

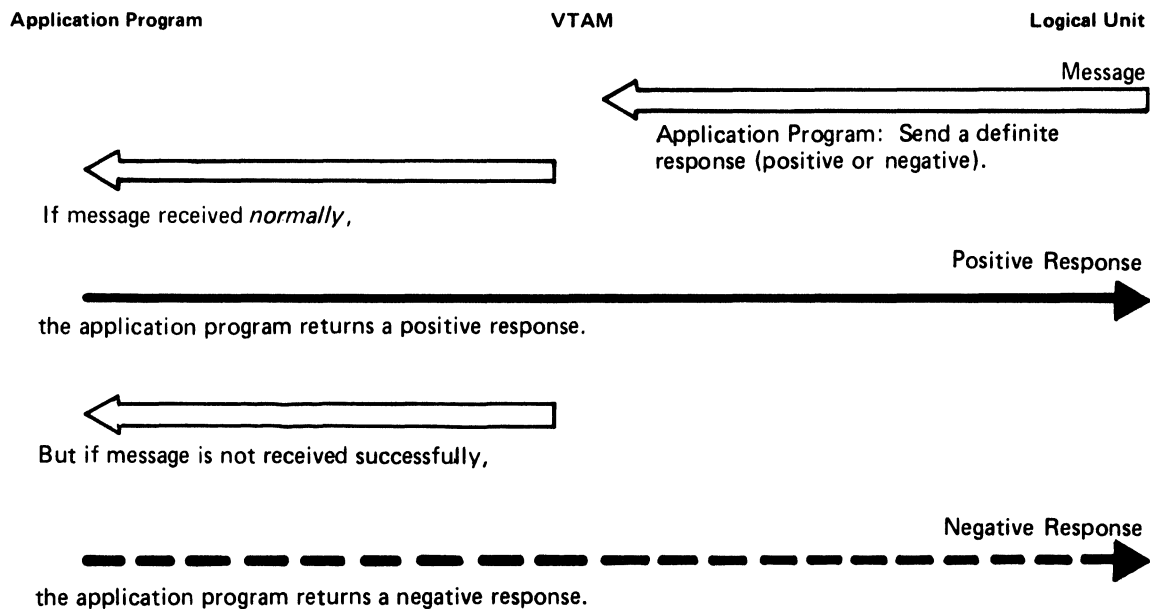
Every response, whether positive or negative, is designated by its sender as a definite response 1 or definite response 2. Many application programs may not need to distinguish between two different kinds of positive and negative responses. For those that do, the distinction may be made for any purpose understood by the user.

If a distinction is made between definite response 1 and 2, it is made as follows: The sender of a message requests that a definite response be returned; it may specify that either definite response 1, definite response 2, or both be returned. The receiver of the message, in returning either a positive or negative response as appropriate, also indicates the corresponding response type (definite response 1, definite response 2, or both) that was requested in the message. The receiver must send back the same type or types of definite responses that were requested.

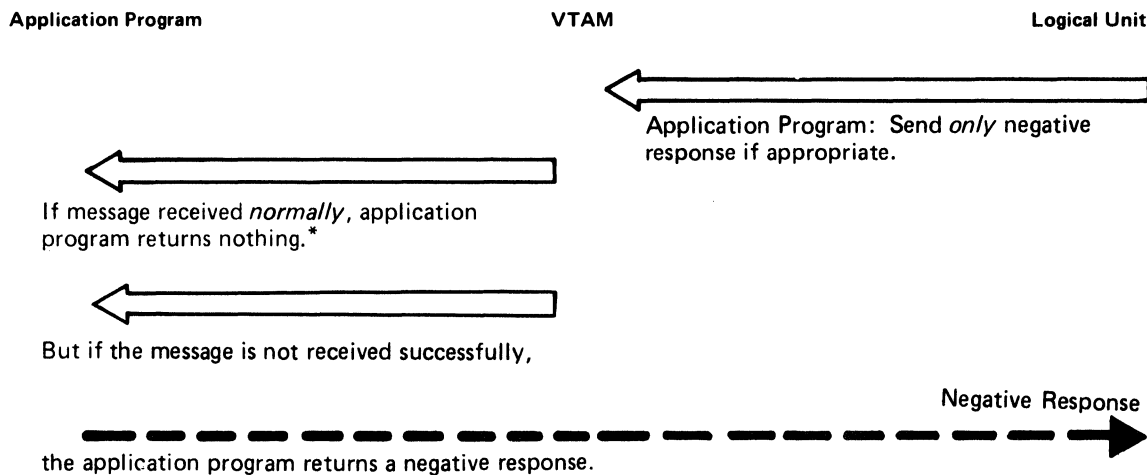
**Sequencing and Chaining**

VTAM assigns a sequence number to each normal-flow message sent to a terminal in record mode. The numbering begins with the first message sent after connection. The number is increased by one for each subsequent message. This process continues until the terminal is disconnected, unless interrupted earlier by the application program. A logical unit assigns sequence numbers to each normal-flow message sent to an application program.

### A Logical Unit Requests A Definite Response



### B Logical Unit Requests Only An Exception Response



\* Even though a message arrives normally (as far as VTAM is concerned), the application program can determine for its own reasons that the message is "defective" and return a negative response, rather than a positive response or no response.

Figure 5-8. An Example of a Logical Unit Requesting (A) Definite Response and (B) Only an Exception Response

Should a message arrive out of sequence (that is, its sequence number is not one greater than that of the last one received), VTAM considers this a transmission error and replaces the message with a message that indicates the error.

When a response is sent (either positive or negative), the sender assigns to it the sequence number of the message being responded to. This provides the sender of a message with a means of matching the response with its message. For example, an application program might send a group of messages, with each message indicating that only negative responses should be returned. Should a negative response be returned, the application program can use the sequence number to determine where in the group the error occurred. Sequence numbers are also useful for terminals that log each message received or sent.

Application programs or terminals can group any number of messages into a set called a message *chain*. The sender can indicate which part of a chain is being transmitted—the first message of the chain, the last message of the chain, the middle of the chain, or both the beginning and end of a chain (the message is the sole element of the chain).

The sender of a chain can at any time send a *Cancel command* to the receiver. The sender might send this command because a negative response was returned for one of the messages in the chain. The receiver can interpret this as an indication that all received messages of the current chain should be discarded.

The actual unit of work that the chain represents is determined entirely by the application program and the terminal. Figure 5-9 illustrates a possible use of chaining. In this example, a terminal has submitted an inquiry to the application program. The application program can obtain various pieces of information from data files and send them to the terminal as each becomes available. By chaining the output requests, the application program has a convenient way of telling the terminal whether any given piece of data represents the beginning, middle, or end of a reply to an inquiry and of checking all the reply before displaying any part of it.

#### **Sending Messages: Scheduled and Responded Output**

Messages can be sent to a terminal with one of two options:

**Scheduled output:** The application program indicates that as soon as the message has been scheduled for transmission and the output data area is free, VTAM is to consider the output operation completed (by posting an ECB or scheduling an RPL exit routine and returns control). Scheduled output is illustrated in Figure 5-10.

**Responded output:** The application program indicates that VTAM is not to consider the operation completed until the message has been received by the terminal and a response has been returned. Responded output is illustrated in Figure 5-11.

These options define the completion of an output operation, and should not be confused with synchronous and asynchronous request handling, which indicate the action to be taken when the completion occurs.

Responded output requires that the output data area and RPL not be reused until a response has been received. If the response indicates that an error occurred, the data is still available for retransmission. Scheduled output allows the application program to send a series of messages that all use the same I/O area and RPL.

With responded output, completion status information is returned when the output request is completed. But with scheduled output, the output request is completed before any completion status information is available. To determine how the output operation was completed, the application program must issue a RECEIVE macro instruction to

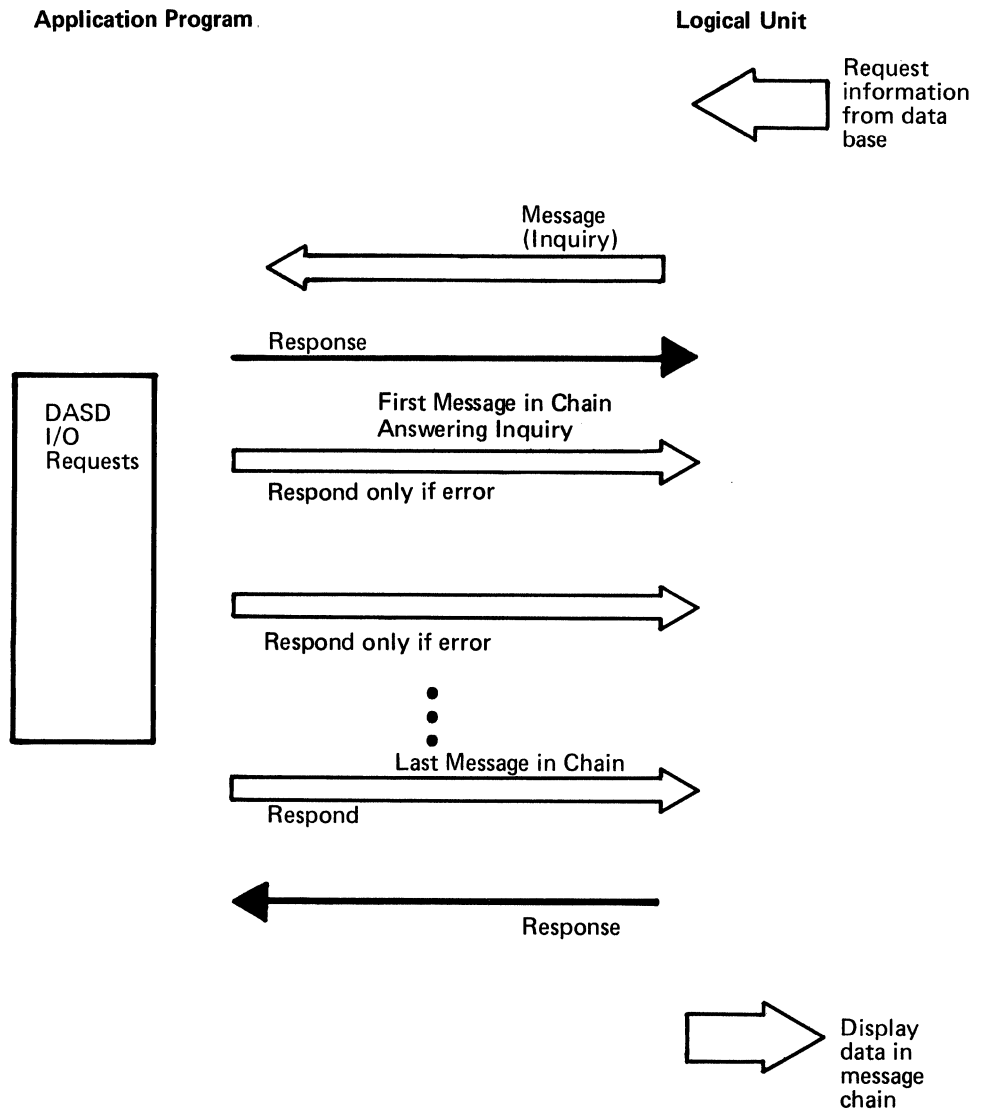


Figure 5-9. An Example of Message Chaining

obtain the completion status information. That is why the application program in Figure 5-10 issues three input requests in addition to the three output requests.

If an error occurs during the transmission of the message, the receiver is passed a substitute message that indicates the error condition. This message is called an *exception message*. The node transmitting the message becomes aware of the error condition when the other node returns a *negative response*.

### Receiving Input

Data, responses, and data flow control information can be received by the application program separately or with one RECEIVE macro instruction. When the macro instruction is issued, any combination of the following types of input can satisfy the request (any combination can be specified):

- Normal-flow messages
- Expedited-flow messages
- Responses



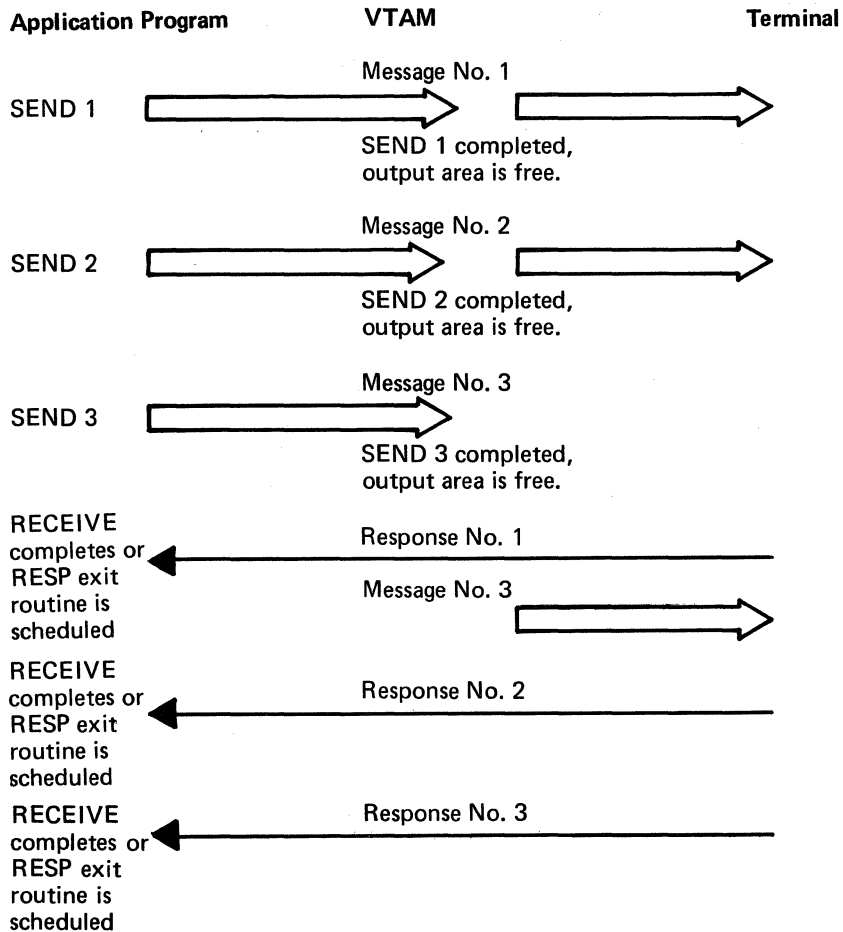


Figure 5-10. Scheduled Output

Normal- and expedited-flow messages are terms used to group all messages into two distinct types of input. Two types of input results are necessary due to these characteristics of data transmission:

If messages are sent at a faster rate than the receiver receives them, the messages are queued for the receiver.

Some messages should not be queued with the other messages but should be available to the receiver separately and immediately.

Thus, VTAM provides two priorities for messages. Data messages are always treated as normal-flow messages. Certain flow-control information is treated the same way. One example is the Quiesce Complete command, which must keep its place in the queue of data messages; if it were to be received prematurely, the bypassed data messages might be lost.

Other flow-control information must be made available immediately to the receiver and is therefore sent to the receiver as expedited-flow messages. One example of an expedited-flow message is the Quiesce at End of Chain (QEC) command. This type of message is not meant to stay within a queue of data messages, waiting until the receiver eventually obtains it.

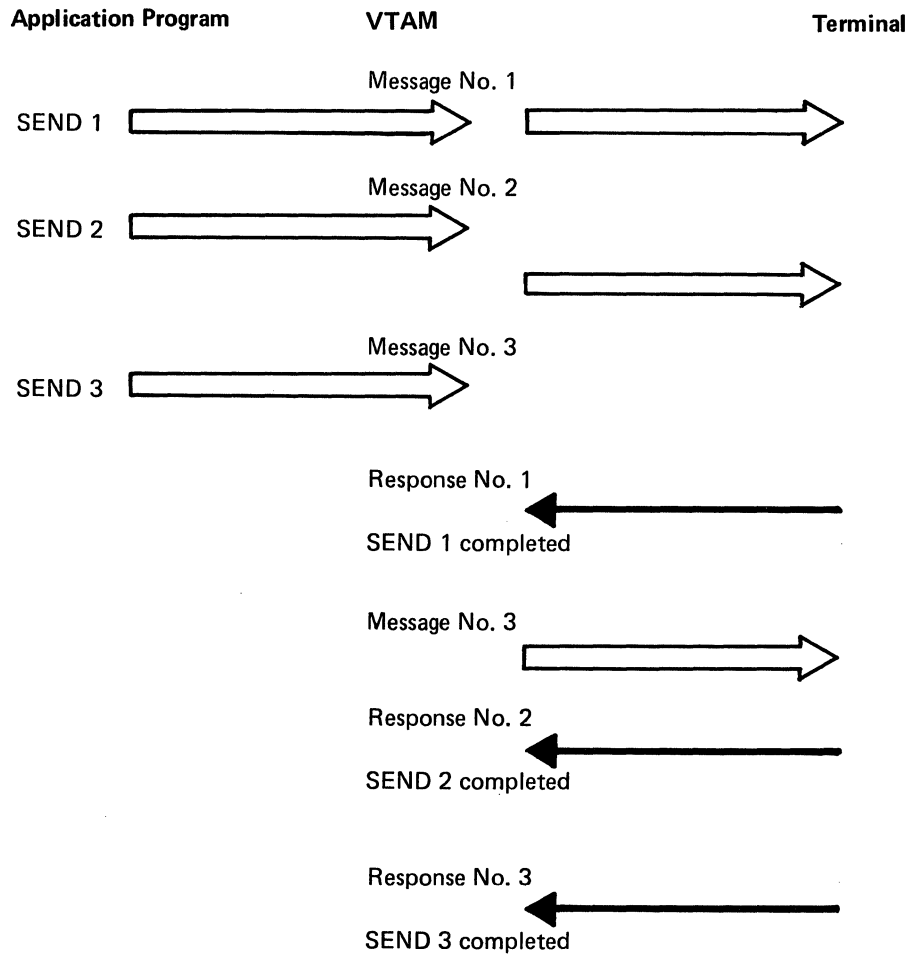


Figure 5-11. Responded Output

Figure 5-12 shows the types of information exchanged between an application program and terminals. See Appendix C to determine which messages are normal-flow and which are expedited-flow.

The application program can maintain three types of exit routines that VTAM schedules whenever one of the following types of input becomes available:

- Expedited-flow messages (DFASY exit routine)
- Responses (RESP exit routine)
- Request Recovery (RQR) commands (SCIP exit routine)

Except for the SESSIONC commands, the type of input that can cause the exit routine to be scheduled corresponds to the type of input that causes a particular type of input request to be completed. Unless an SCIP exit routine is maintained, the application program has no means of receiving an RQR command.

These exit routines operate in the same manner as those described in “EXLST Exit Routines,” earlier in this chapter. One difference, however, is that the application program need not use one exit routine to handle a particular kind of input from all terminals. A given exit routine can service input from a limited set of terminals, or a separate exit routine can be written for each terminal.

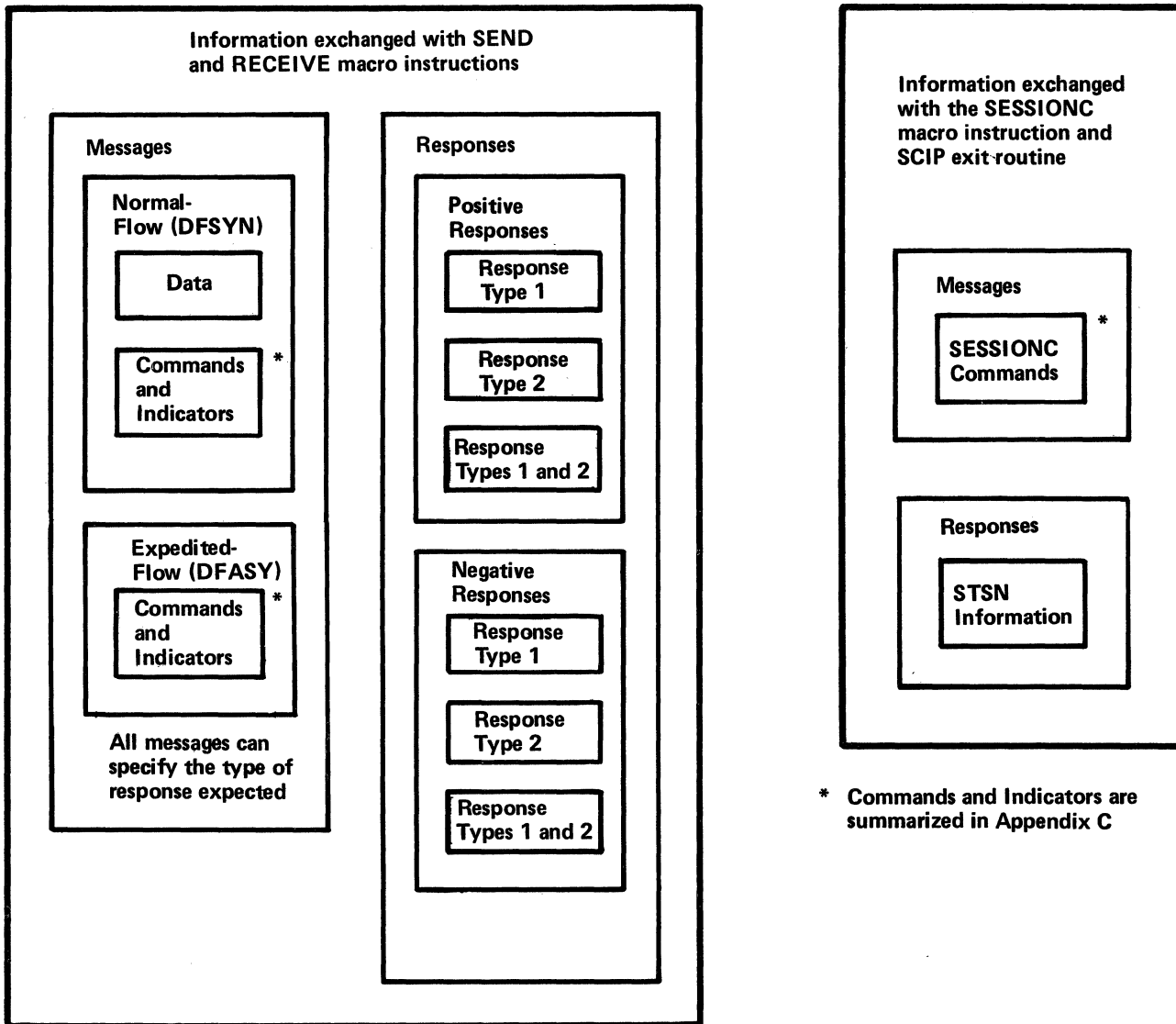
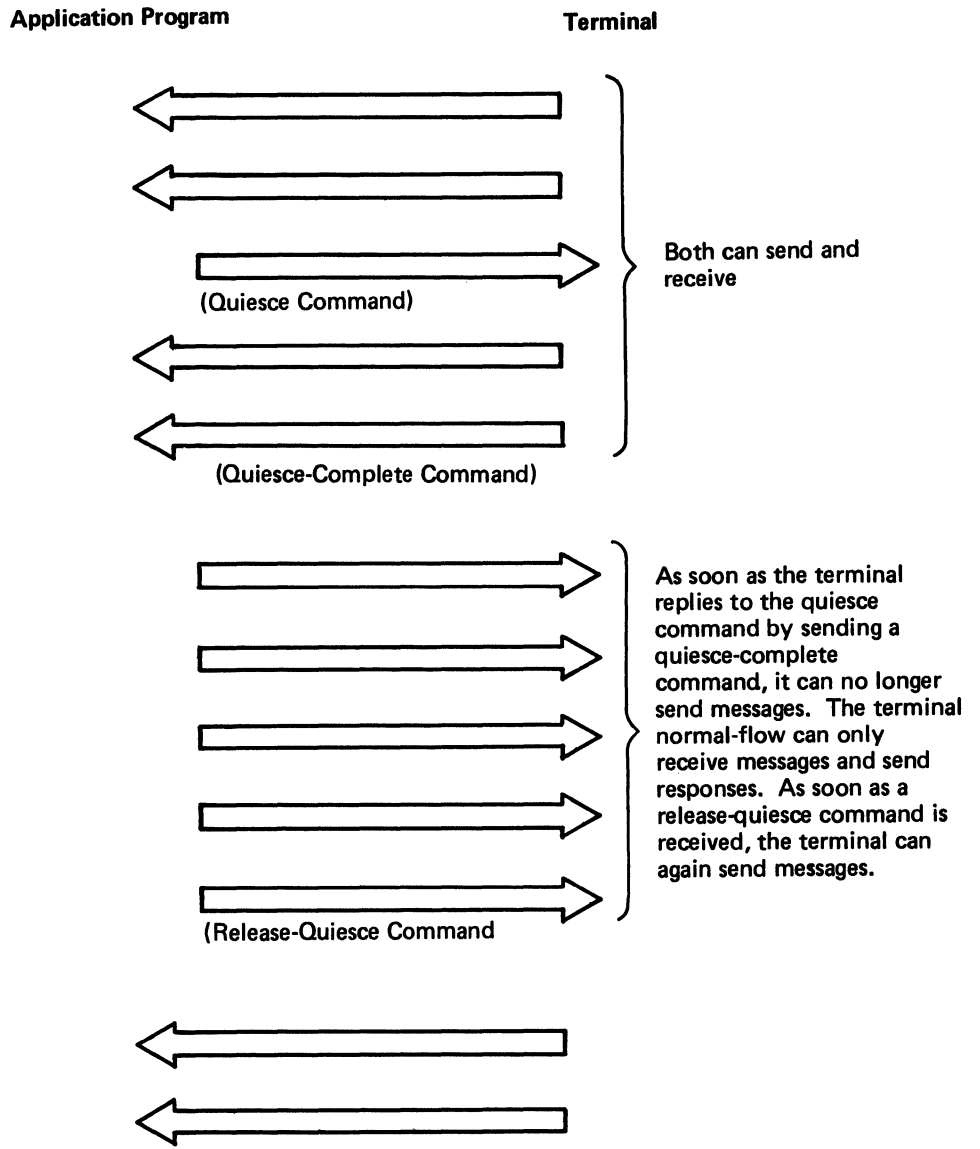


Figure 5-12. Types of Information Exchanged Between an Application Program and Terminals

**Quiesce, Change-Direction, and Bracket Protocols**

Messages can be sent to a terminal whether or not it is at that moment sending messages to the application program. The nature of some application programs, however, may prohibit such unrestricted exchange of data, or the application program may have been written when no capability for unrestricted data exchange existed, and the user does not wish to rewrite the program to use this facility. For such application programs, three methods of communication are available that allow the application program and the terminal to control each other's ability to send data. These three methods (which are essentially three sets of commands and indicators with "rules" about how to use them), are called *quiesce protocol*, *change-direction protocol*, and *bracket protocol*.

**Quiesce Protocol:** The application program informs the terminal that it is to stop sending normal-flow messages when it has completed sending its current chain. It does so by sending a *Quiesce at End of Chain (QEC) command*. When the terminal replies, by sending a *Quiesce Complete (QC) reply*, it must stop sending and prepare to receive. The terminal cannot again send data until it receives a *Release Quiesce (RELQ) command* from the application program, as shown in Figure 5-13. In this example, the VTAM



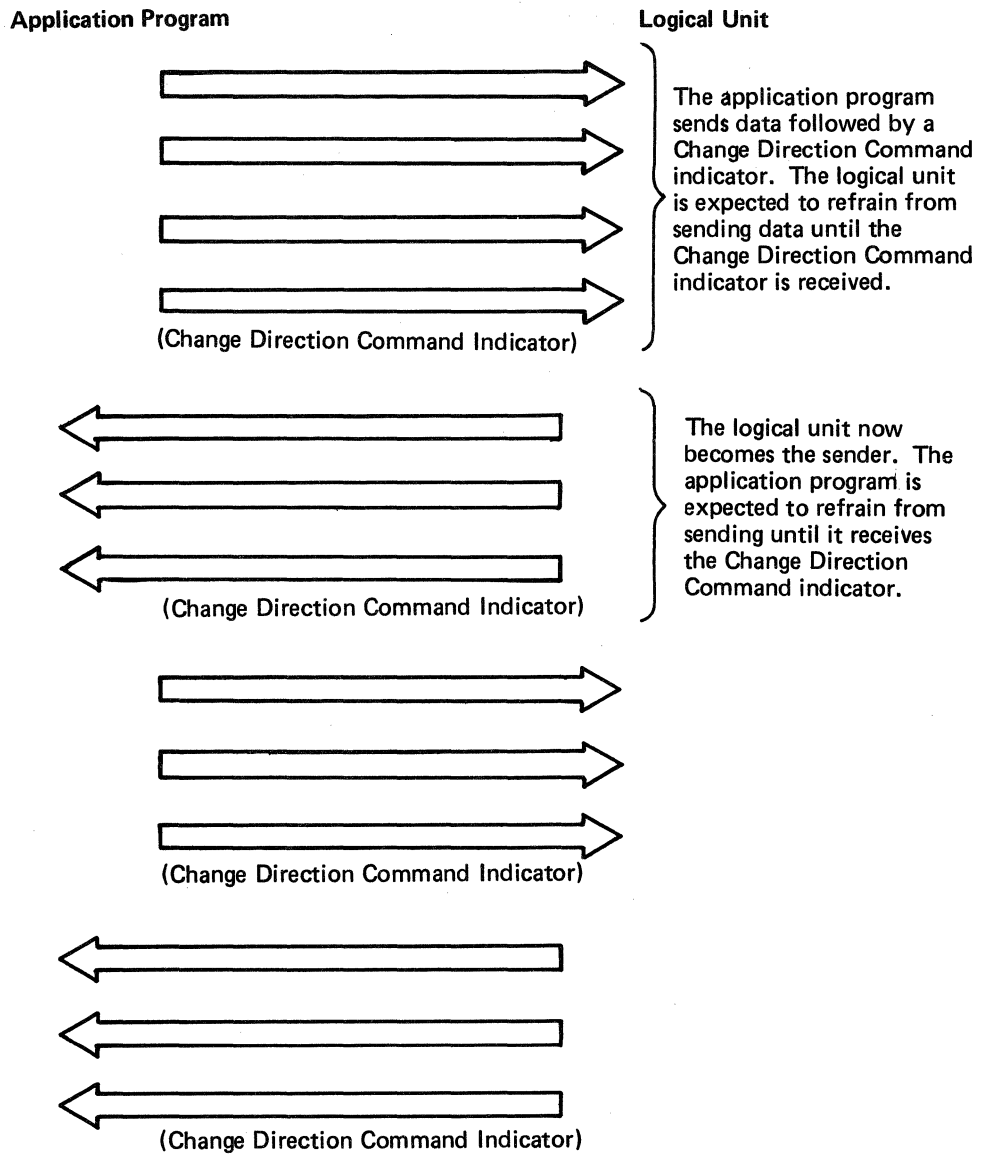
*Note:* Responses are not shown

Figure 5-13. An Example of Quiesce Protocol

application program sends the QEC command, but it can also be sent by a terminal. VTAM does not enforce quiesce protocol; it is the user's responsibility to conform to the convention.

**Change-Direction Protocol:** The first node that sends data (following the application program's indication that data flow can begin) can continue to send data. When the first node is through sending data, it sends a *Change Direction Command indicator* to the other node. The other node sends data until it relinquishes its ability to send by returning another Change Direction Command indicator. The nodes continue to alternate in this fashion, as shown in Figure 5-14.

The node that is awaiting a Change Direction Command indicator, like the node that is in a quiesced state, is free to send responses. Only the sending of normal-flow messages is



Note: Responses are not shown.

Figure 5-14. An Example of Change-Direction Protocol

restricted. While the receiver is awaiting the Change Direction Command indicator, it can transmit (as part of a response) a prompting indicator to the other node that in effect says "I would like that Change Direction Command indicator now." The prompting indicator, called a *Change Direction Request indicator*, can be honored or it can be ignored.

Change-direction protocol is not enforced by VTAM. Should the node waiting for a Change Direction Command indicator begin sending data anyway, VTAM does not prevent the transmission of the data. The successful use of this method of communication rests on the assumption that neither the application program nor the terminal violates the "rules."

**Bracket Protocol:** A *bracket* is any unit of work for which the application program and the terminal have been programmed. Each bracket consists of input operations or output operations (or both) that do not necessarily follow a fixed pattern. Data-base inquiry and data-base update transactions are typical examples of brackets.

Bracket protocol is used when one of the nodes cannot process a new bracket until the previous one has been completed. Nodes using this method of communication note on each transmitted message whether that message is the beginning, middle, or end of a bracket. These delimiters allow the receiving node to determine whether or not a new bracket can be started. Figure 5-15 illustrates a use of bracket protocol.

Because either node can initiate a bracket, a *Bid command* can be used to avoid situations in which both nodes attempt to initiate a bracket at the same time. A Bid command requests permission to start a bracket. Upon receipt of a Bid command, the receiving node can reject it, give permission for a bracket to be initiated, or reject the Bid

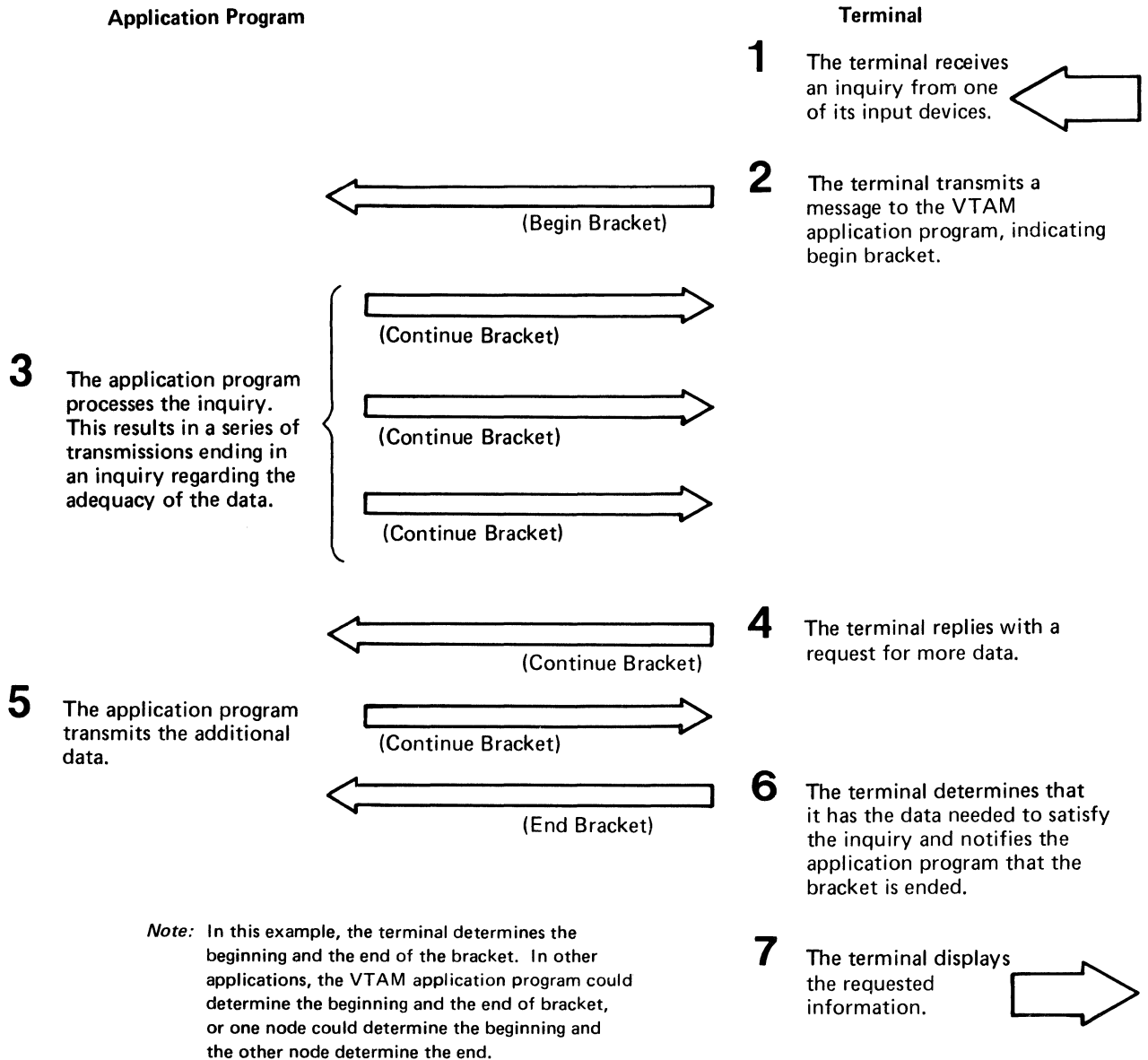


Figure 5-15. An Example of Bracket Protocol

temporarily. If the Bid is rejected temporarily, the node that received it transmits a *Ready to Receive (RTR) command* when a bracket session can be permitted. Upon receipt of an RTR command, the node that originally sent the Bid can initiate the bracket by sending a Begin Bracket indicator in a message.

Only one node is permitted to initiate a bracket without the use of the Bid. The other node is required to use the Bid before initiating a bracket.

Like quiesce protocol and change-direction protocol, bracket protocol is not enforced by VTAM, except for 3270 communication. Any message containing a Begin Bracket indicator that is sent before the previous bracket has ended is not rejected by VTAM.

Figure 5-16 lists the three sets of indicators and commands discussed above.

### Request Mode Selection

When connection is established, an application program and a terminal decide how often a response must be requested and in what order responses must be returned. *Immediate control mode* requires one response for each message sent; *delayed control mode* allows more than one message to be sent before a response is requested.

**Immediate Control Mode:** When immediate control mode is used, the sender sends only single-message chains and requests a definite response for each message sent. No more messages are sent until a definite response is received.

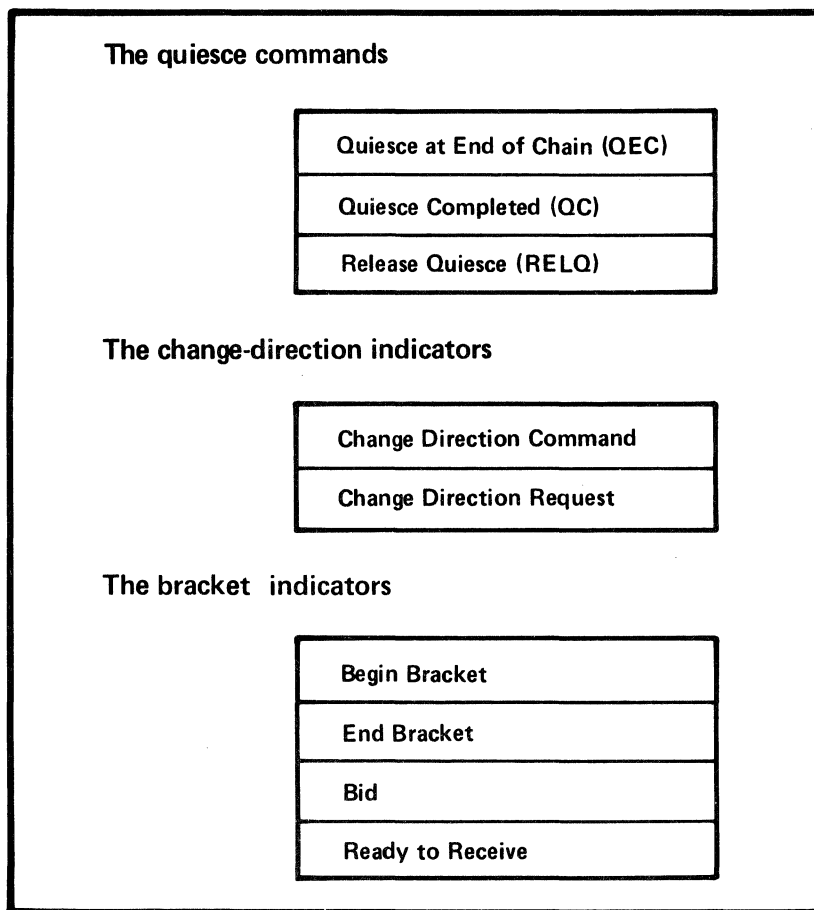


Figure 5-16. Commands and Indicators Used to Direct the Flow of Data

**Delayed Control Mode:** When delayed control mode is used, one or more chains of messages can be sent before requesting a definite response. A definite response is requested in the last message of a chain; all other messages request only exception responses. There are two forms of delayed control mode:

*Immediate request mode* allows one or more chains of messages to be sent before a definite response is requested. No more messages are sent until a definite response is received. The receiver must return responses in the same order as they are requested.

*Delayed request mode* allows one or more chains of messages to be sent before a definite response is requested. The sender need not wait for a response before sending more chains of messages. The receiver may return responses in any order.

#### **Additional Protocols**

In addition to the protocols described in this chapter, the following protocols can be specified during a connection:

Whether or not a terminal can remove extraneous blank characters before data is transmitted (compression)

Who has error recovery responsibility

Whether an alternate character code is acceptable

Whether communication is simplex, half-duplex, or duplex

#### **Communicating with the 3270 Information Display System in Record Mode**

A VTAM application program can communicate with the 3270 Information Display System in record mode. That is, the application program uses SEND and RECEIVE macro instructions.

The following restrictions apply to communication with a local, BSC, or SDLC (SNA) 3270 (all other aspects of record-mode communication apply as described previously):

- All commands and orders for the 3270 must be placed in the output data by the application program. Data, therefore, includes 3270 commands and orders.
- No responses should be sent to a 3270. All incoming messages indicate that no response of any type is expected.
- Messages sent to a 3270 can contain only data (including 3270 commands and orders). No quiesce, Bid, Chase, or Cancel commands and no change-direction indicators should be sent. Bracket indicators can be set, but chaining indicators should always mark the message as the sole element of a chain (all incoming messages are so marked).
- No SESSIONC commands can be received from a 3270. Only the Clear command can be sent to it. The effect of the Clear command is to reset both incoming and outgoing sequence numbers to 0.
- The bracket convention must be used. If the application program has no use for brackets, the entire interval between the first I/O operation of a connection and disconnection can be considered to be one bracket. Both the application program and the 3270 can begin a bracket.

The first input from a 3270 is marked as the beginning of a bracket. All subsequent messages received from the 3270 during the session indicate that the bracket is being continued. The 3270 cannot end a bracket; this can only be done by the application program.

- The application program should request a definite response 1 for each message sent to the 3270 that begins or ends a bracket.



Chapter 8 discusses additional restrictions that apply to local and BSC terminals. Appendix D discusses compatibility considerations for communicating so that a program can communicate with both BSC and local 3270 terminals and SNA 3270 terminals in the same way or so that BSC 3270 terminals can be replaced with SNA 3270 terminals with a minimum amount of difficulty. See the chapter on VTAM support in *Introduction to Programming the IBM 3270* for a summary of how to use VTAM with the 3270.

### ***Basic-Mode and Record-Mode Communication***

Identification of terminals, use of specific-mode and any-mode I/O requests, use of continue-specific and continue-any modes, and handling of excess data are the same for basic-mode communication and record-mode communication. This section should be used in conjunction with “Record-Mode Communication” in this chapter and “Basic-Mode Concepts” in Chapter 8.

### **Identifying Terminals**

Before an application program is connected to a terminal, it refers to the terminal by means of the terminal’s 8-byte symbolic name created during VTAM definition. When connection is established with the terminal during program execution, VTAM provides the application program with a network-oriented name, called a *communication identifier* or *CID* for that connection. The application program uses the CID for all I/O requests issued in specific-mode.

When a RECEIVE macro instruction issued in any-mode is completed, VTAM provides the CID of the terminal that sent the data. Because the application program will probably communicate with the terminal in specific-mode, VTAM supplies the CID, rather than the symbolic name, to the application program. Should the identity be significant, the application program has three ways to relate the CID to the terminal’s symbolic name:

The application program can ask VTAM to translate the CID into the terminal’s symbolic name using the INQUIRE macro instruction.

The application program can maintain a table of CIDs and their equivalent symbolic names.

When the application program establishes connection with the terminal, the application program can initially assign a 4-byte value that VTAM returns each time that terminal’s data satisfies a RECEIVE. The 4-byte value can be anything the application program chooses to associate with the terminal; it occupies the user fields of the RPL (USERID and USER). It can be used to identify the terminal or it could contain the address of a subroutine that is to handle that terminal’s data.

### **Specific-Mode and Any-Mode**

To obtain data from a terminal, the application program can request data from a *specific* terminal, or it can request data from *any* one of its connected terminals. The application program designates the desired mode—specific or any—with each READ, SOLICIT, or RECEIVE macro instruction. These two modes are called, respectively, *specific-mode* and *any-mode*.

In general, an application program initially requests input in any-mode, and then communicates with each terminal in specific-mode until the transaction, inquiry, or conversation is completed. While communication in specific-mode is taking place with one terminal, a RECEIVE, READ, or SOLICIT macro instruction issued in any-mode is pending so that new transactions can be handled.

In any-mode, the application program does not know the identity of the source terminal until the data has been moved into its input area and the READ, SOLICIT, or RECEIVE has been completed. Because the terminal is initially unknown, the amount of incoming data may also be unknown. This means that the application program must either reserve

an input area large enough to hold the largest possible amount of incoming data or execute additional instructions to handle excess data. On the other hand, the any-mode allows the application program to use just one input area for data from all of its terminals rather than using a separate input area for each of its terminals.

With specific-mode, the application program must specify the identity of the terminal supplying the data. Since the identity of the source is known, the size of the input area is much more predictable than with any-mode. But a terminal may not supply data for some time, so the application program may have to contend with unused data areas.

Input data areas can be managed by a combination of specific-mode and any-mode. As an example, consider an application program that obtains an inquiry from any of its terminals, handles that inquiry, and then obtains a new inquiry. Part of such a program for record-mode terminals is illustrated in Figure 5-17.

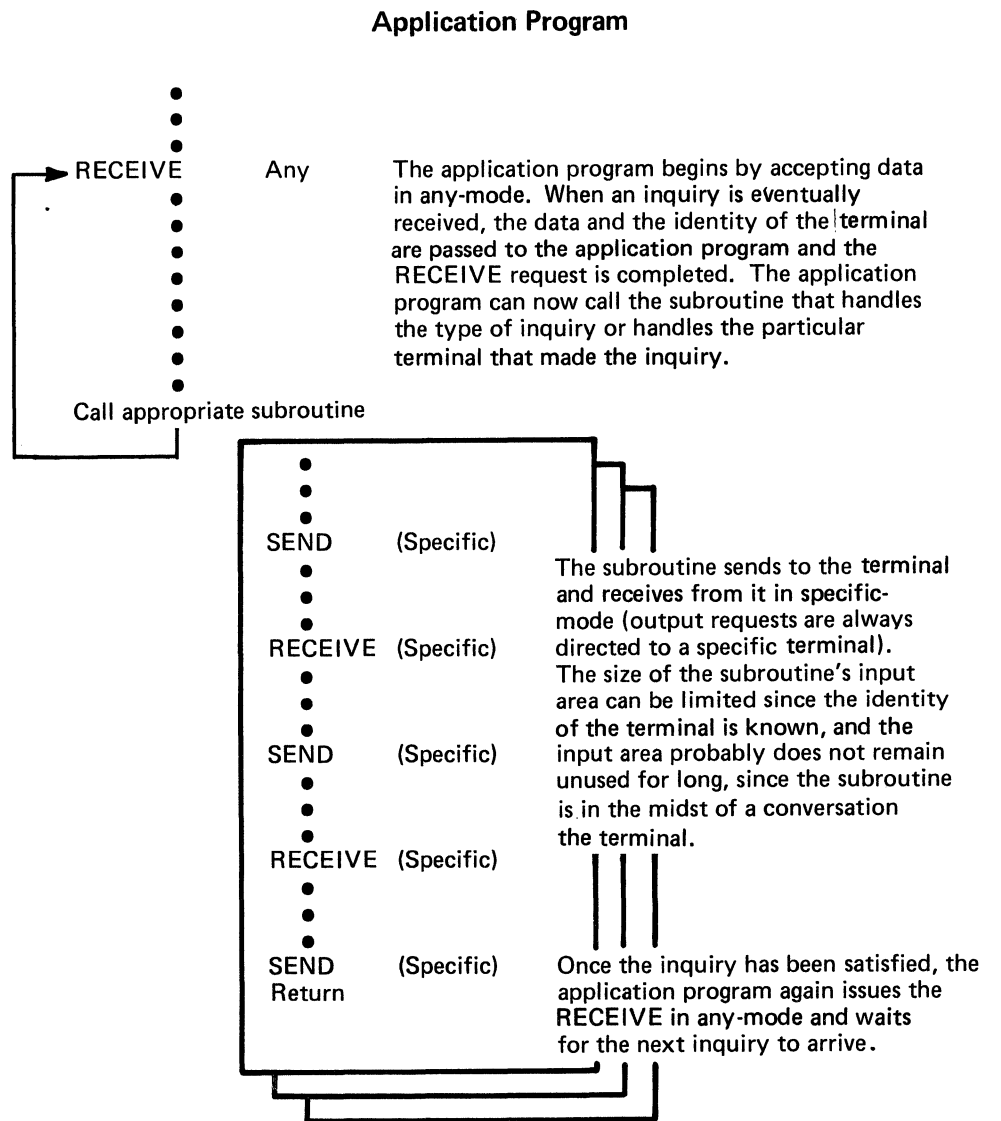


Figure 5-17. Using a Combination of Any-Mode and Specific-Mode to Obtain Data

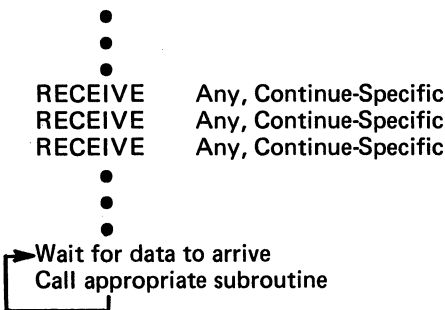
**Continue-Any and Continue-Specific Modes**

In the example of Figure 5-18, synchronous request handling for the I/O requests is assumed. The application program handles each inquiry serially, never accepting a new inquiry until it has processed the previous one. Although this procedure might be suitable for application programs processing short inquiries from a few terminals, most application programs would probably work more efficiently if the inquiries were handled in parallel.

An application program designed to handle more than one inquiry at a time might use asynchronous request handling and issue new READ, SOLICIT, or RECEIVE macro instructions in any-mode before the previous inquiry has been processed. This, however, raises the possibility that both a specific-mode request and an any-mode request might be awaiting data at the same time. Consequently, data for the specific-mode request might instead be intercepted by the any-mode request, which is meant only to receive new inquiries.

To eliminate this sort of problem, VTAM allows the application program to indicate when a READ, SOLICIT, or RECEIVE macro instruction issued in any-mode can receive data from a particular terminal and when a READ, SOLICIT, or RECEIVE macro instruction issued in specific-mode must receive the data. The former is called *continue-any mode*, and the latter is called *continue-specific mode*. These modes are designated when an I/O request is issued, but do not become effective until the I/O

**Application Program**



The application program begins by issuing three RECEIVES in any-mode. Continue-specific mode is also designated for each one; this means that once a terminal sends data and causes one of the RECEIVES to be completed, subsequent data from that terminal can only be obtained with RECEIVES issued in specific-mode.

When the data arrives, the appropriate subroutine determines if the inquiry is completed. If it is not, the subroutine exchanges data in specific-mode. The terminal is kept in continue-specific mode so that the arriving data can only satisfy the RECEIVE issued in specific-mode, not one of the RECEIVES issued in any-mode.

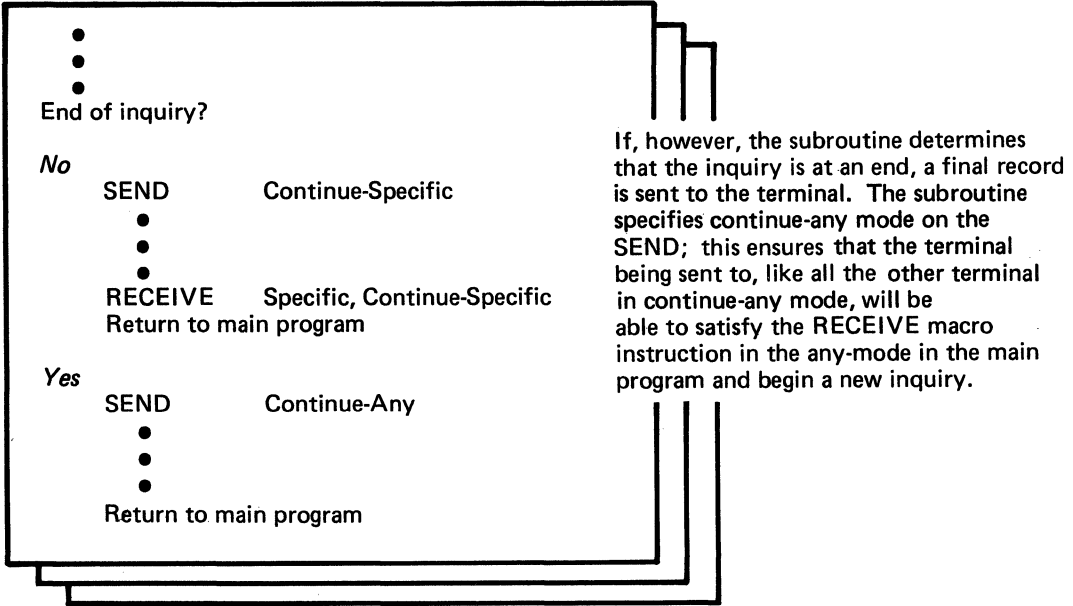


Figure 5-18. Using the Continue-Any and Continue-Specific Modes to Handle Concurrent Inquiries

operation is completed. Alternatively, a terminal can, at connection, be designated as always in either continue-any or continue-specific mode. If the terminal is always in one of these modes, the mode specified in the I/O request is disregarded. This facility makes it possible to treat terminals that are expected to always enter one-line input (always in continue-any mode) differently from those that may enter multiple-lines of input (only the first line received in continue-any mode).

Figure 5-18 illustrates how the modes described above are related. Although Figure 5-18 shows record-mode communication using continue-any and continue-specific modes, the same relationship exists for basic-mode communication.

### **Handling Excess Input Data**

When an application program issues a READ, SOLICIT, or RECEIVE macro instruction, the length of the incoming data is often unpredictable. As noted above, this is particularly true of RECEIVE macro instructions issued in any-mode. VTAM provides two ways of handling data that is too large for the input area:

*VTAM can discard the excess data.* This facility is called the TRUNC (truncate) option. This option would be useful in application programs that must impose rigid size limitations on input data. For example, an inventory-control application program might require the terminal to supply an account number no longer than 10 bytes.

*VTAM can keep the data.* VTAM fills the input area, saves the remainder, and completes the input request. Additional input requests must be issued to obtain the excess data. This facility is called the KEEP option.

The application program selects the appropriate option when a terminal is connected, but it can override the option for a given request.



## CHAPTER 6. RELIABILITY, AVAILABILITY, SERVICEABILITY

This chapter describes how VTAM detects errors, attempts to recover from them, and—when recovery is impossible—records the conditions under which the error occurred. The chapter also describes VTAM debugging aids, dump programs, and test programs used to correct programming errors and machine malfunctions.

### VTAM's RAS Strategy

VTAM's reliability, availability, and serviceability (RAS) aids are intended to minimize the impact of programming errors and machine malfunctions on the telecommunication network. VTAM attempts to detect problems and handle them before they become serious. If an error is encountered, VTAM attempts recovery and records the error for possible future maintenance. If recovery is not possible, VTAM automatically provides error records and dumps to help identify the problem and its cause. For additional debugging aid, the user can request traces and testing services from VTAM. If possible, VTAM avoids terminating application programs or closing down the telecommunication system when an error occurs; instead, it tries to isolate the problem and either correct the error itself or provide information to enable the user to correct it. VTAM handles both hardware and software errors; software errors include those on the part of users (application programs and the network operator), of the operating system, and of VTAM itself.

Error-recovery and error-recording operations are invoked automatically. VTAM requires no action on the part of the user to invoke them, although the error records can be used to perform maintenance on elements in the system that are causing frequent temporary errors.

When an error is permanent, a message is usually sent to the network operator. The message and its explanation define the condition, indicate the probable cause, and suggest a course of action. This action may involve circumventing the problem as well as collecting data for later debugging. To get adequate material for problem determination, the user does one or more of the following things:

- Saves all console logs that pertain to the error message. These logs probably reflect all of VTAM's actions from VTAM startup through the issuing of the message.
- Obtains printouts of any traces performed in connection with the error.
- Saves any dumps that may have resulted from the error.
- Requests and saves status displays of the nodes involved in the error.
- Initiates the Teleprocessing Online Test Executive Program (TOLTEP) for testing devices and lines involved in the error.

As a temporary solution to a problem, the network operator may be able to circumvent it or avoid it by disabling part or all of the telecommunication system. In any case, to assist in later correction of the problem, the user should keep a complete description of the network as it existed at the time of the error.

Correcting the problem involves identifying the cause and then making the necessary correction. The following steps may help identify the problem:

1. Studying the message log: The sequence of messages leading up to the error as well as the error message itself may identify the problem.

2. Examining error records: Error records pertaining to VTAM or its telecommunication system may identify the problem.
3. Examining dumps and traces: Dumps and traces can be used to identify the area in which the error occurred or to find the problem itself.
4. Re-creating the problem: VTAM's network operator commands and RAS facilities can help re-create the problem and to collect pertinent data.

## VTAM's RAS Facilities

VTAM's RAS facilities can be grouped into those related to serviceability and those related to reliability and availability of the system. The following sections detail VTAM's RAS facilities.

### *Serviceability Aids*

VTAM's serviceability aids assist in determining the cause of problems in the telecommunication system. Serviceability aids include:

#### **Error Recording**

*Hardware Errors:* VTAM's hardware-error recording augments the error recording of the operating system. Both permanent and temporary errors are recorded. In addition, messages are sent to the network operator when any permanent error occurs.

*Software Errors (OS/VS):* VTAM uses SYS1.LOGREC to retain records of software errors that result in the abnormal termination of VTAM.

#### **Traces**

*DOS/VS:* The problem determination and serviceability aids (PDAIDS) can be used to monitor VTAM's SVCs, I/O operations, and internal storage management.

*OS/VS:* The generalized trace facility (GTF) can be used to monitor VTAM's SVCs, I/O operations, and task management and to trace events in application programs using VTAM.

*VTAM:* Internal VTAM traces can be used to monitor I/O activity, to record the contents of VTAM buffers, and in OS/VS, to trace VTAM storage management.

#### **Dumps**

*OS/VS:* System dump programs are tailored by VTAM to provide formatted dumps of some VTAM control blocks and in some cases to provide additional diagnostic information.

*DOS/VS:* VTAM provides no dump facilities for DOS/VS. However, system dump programs can be used to dump some VTAM control blocks and data areas.

*NCP:* VTAM has tailored the NCP dump utility program. This utility program can be invoked using VTAM's MODIFY command, or, optionally, it can be invoked as part of VTAM's error recovery procedures for communications controllers.

#### **TOLTEP**

The Teleprocessing Online Test Executive Program (TOLTEP) provides online testing for communication lines and certain devices in a VTAM system.

### **Error Recording**

VTAM provides facilities for recording hardware errors and software errors.

**Recording Hardware Errors:** Recording hardware errors are a function of the error recovery procedures (ERPs). (See "Reliability and Availability Support," later in this chapter, for a description of ERP processing.) The data on hardware errors collected by VTAM is placed in the error-record data set of the operating system. The information in

this data set can be formatted and printed by each system's environmental recording, editing, and printing (EREP) program.

VTAM's hardware error recording is an extension of the OS/VS outboard recorder and the DOS/VS recovery management support recorder.

In addition to recording errors, VTAM maintains two counters for each local device in the network. One counter keeps track of the number of SIO commands issued for the device; the other keeps track of the number of temporary errors for the device. Counters are also kept in the device statistic tables that the operating system maintains for each local device. These counters indicate the number of each type of error encountered for each device. The tallies in the counters are included in any records written to the error-record data set.

Recording occurs when any of the following conditions is encountered:

*Permanent hardware error:* A record is written when ever there is an error for which recovery could not be made, either because the error is undefined or because attempts by ERPs to correct the problem were unsuccessful. Records for permanent errors include the name and address of the failing device, the time and date of the failure, the contents of the counters at the time of the failure, the failing channel command word, the channel status word, sense information, and device flags.

*Counter overflow:* A record is written whenever any of the counters is about to overflow. Records for counter overflow and end-of-day conditions include the name and address of the associated device, the time and date that the event occurred, and the contents of the counter.

*End-of-day:* A record is written whenever VTAM deactivates a device. Records for counter overflow and end-of-day conditions include the name and address of the associated device, the time and date that the event occurred, and the contents of the counter.

**Recording Software Errors:** In OS/VS systems, if VTAM cannot recover from a software error and must terminate processing, VTAM collects data on the error. This data is formatted and written to the system data set SYS1.LOGREC. Records on this data set can be formatted and printed by the OS/VS EREP program.

A VTAM software error record includes a description of the error and an audit trail. A VTAM audit trail is a record of the modules entered during the processing in which the error occurred. (An audit of module activity is maintained for those areas of VTAM responsible for processing telecommunication requests.) The audit information in the software error record identifies the module being executed when the error was detected, as well as the modules that were entered prior to the error.

## Traces

VTAM under both DOS/VS and OS/VS has operating system traces and VTAM traces. The operating system traces provide the highest level of problem determination. By using the appropriate system trace, a problem may be isolated to a particular program or operating system component. If the problem is isolated to a component such as VTAM or to an application program using VTAM, the VTAM traces can then be used to help locate and identify the area in which the error is occurring.

**Operating-System Traces:** The system traces can be used to monitor the SVC and the I/O activities of VTAM. The data collected by these traces can be used as a chronology of VTAM activity and reflects the conditions in VTAM during errors. System traces for VTAM are started and stopped by using the facilities of the operating system.



DOS/VS trace records for VTAM are printed with the PDLIST program. OS/VS1 trace records for VTAM are printed with the operating system's HMDPRDMP service aid. Refer to the publication *OS/VS1 Service Aids*, for information on HMDPRDMP. OS/VS2 trace records for VTAM are printed using the operating system's AMDPRDMP service aid. Refer to the publication *OS/VS2 System Programming Library: Service Aids*, for information on AMDPRDMP.

**VTAM Traces:** In addition to the operating system traces, the following types of VTAM traces are provided:

*I/O traces:* To trace I/O activity within VTAM.

*Buffer traces:* To record the contents of VTAM buffers as data enters and leaves VTAM.

*NCP line traces:* To record the NCP trace records of specified line.

*Storage management traces:* To record the use of VTAM storage.

VTAM traces are initiated and terminated by VTAM start options or the MODIFY command. See "VTAM Commands" in Chapter 4 for information on starting and stopping VTAM traces.

The VTAM trace records are written on a particular data set in each operating system. For DOS/VS, the file name of this data set is TRFILE. For OS/VS, VTAM writes the VTAM trace records for all traces except the storage management traces. The system PDAIDs need only be active when the storage management traces are used. For OS/VS, VTAM passes the trace data to GTF, which records it in SYS1.TRACE. Because VTAM traces use GTF to record the trace records, GTF must be active before VTAM data can be recorded. If GTF is not active, VTAM does not provide a record of the event to be traced.

To print VTAM trace records, VTAM provides a utility program for DOS/VS and uses a system utility program for OS/VS. The VTAM trace-print utility program for DOS/VS selectively edits and prints the data collected by the various VTAM traces. Using the MODIFY command, the network operator starts the print utility program and specifies the nodes for which trace records are to be printed. In OS/VS1, VTAM traces are printed by using the operating system's HMDPRDMP service aid. In OS/VS2, VTAM traces are printed by using the operating system's AMDPRDMP service aid.

Refer to the publication *DOS/VS Serviceability Aids and Debugging Procedures*, for more information on the PDAIDs and the PDLIST Program. Refer to the publication *OS/VS1 Service Aids*, for more information on GTF and the HMDPRDMP service aid. Refer to the publication *OS/VS2 System Programming Library: Service Aids*, for more information on GTF and the AMDPRDMP service aid.

## Dumps

VTAM provides routines to augment both the operating system dump facilities and the NCP dump facilities. The text below describes what these routines do.

**Operating System Dumps:** VTAM provides routines that format portions of OS/VS dumps. When the operating system is dumping VTAM or an application program that is using VTAM, it invokes these routines to locate and format key VTAM control blocks. For each control block that is formatted, its name and address is printed, followed by the name and contents of the pertinent fields. A hexadecimal dump of the control block is printed following the formatted printout. If VTAM finds an invalid field (either in the control block or in the chain of pointers to the control block), the condition is noted in the dump.

This formatting is performed automatically for dumps resulting from abnormal termination (ABEND) of VTAM itself, from the abnormal termination of an application

program that is using VTAM, or from a SNAP macro instruction in an application program. For dumps produced by the PRDMP service aid, however, VTAM formatting is optional; the option must be specified as part of the input to the service aid.

Dumps of VTAM include an audit trail and unique identifiers for each VTAM module and control block. The audit trail is like that recorded for software errors.

**NCP Dumps:** VTAM uses the NCP dump program to dump the contents of communications controllers. An NCP dump is taken automatically if the NCP fails and an automatic dump was specified as part of the generation of that NCP. If an NCP fails and an automatic dump was not specified, the network operator is notified of the failure and is given the option of requesting a dump of the NCP. Other than during error recovery, the network operator can also request a dump of an NCP by using the MODIFY command, as described in “Starting and Stopping VTAM Facilities” in Chapter 4.

**Teleprocessing Online Test  
Executive Program  
(TOLTEP)**

TOLTEP is a VTAM component that controls the selection, loading, and execution of online tests (OLTs) within the telecommunication network. These OLTs are specific device tests designed to be used to diagnose hardware problems and to verify the reliability of a device in the telecommunication network.

TOLTEP allows multiple OLTs to be run concurrently with application programs in the VTAM telecommunication system. TOLTEP supports testing of devices attached on start-stop or BSC lines, devices in the 3270 family, SDLC lines, and the 3767 and 3770 family. In OS/VS, TOLTEP can also be used to dynamically modify its configuration data sets (CDS files).

TOLTEP is included automatically in the system when VTAM is generated. It requires that the OLT option be specified during NCP generation. TOLTEP is automatically initialized and made ready to accept test requests when VTAM is started. If TOLTEP is abnormally terminated prior to VTAM termination, VTAM automatically attempts to restart it.

To run TOLTEP, a terminal must be available for use as a control terminal. This terminal can be the terminal to be tested, but usually is not. The control terminal must be able to enter and receive alphanumeric characters. An alternate printer can be designated to receive hard-copy output from the test. (The alternate printer is required if the control terminal is also the terminal being tested.)

TOLTEP can be invoked either by the network operator or by an operator at a network terminal. As explained in “Starting and Stopping VTAM Facilities” in Chapter 4 TOLTEP can be invoked by the network operator by an option of the MODIFY command or by the VARY command to log a specific terminal on to TOLTEP.

If a terminal is being monitored by the network solicitor (and is therefore not connected to an application program), a terminal-initiated logon can be used to log the terminal on to TOLTEP.

If a terminal to be used by TOLTEP is currently connected to an application program, the connection must be broken before the terminal can be made available to TOLTEP. VTAM allows an application program to be notified if a connected terminal is to be used by TOLTEP; the application program must issue a CLSDST macro instruction to release the terminal. If a connected terminal specified in a test request is not released by the application program, TOLTEP waits until the terminal is available before proceeding with the test.

When TOLTEP is invoked, it requests information from the operator at the invoking terminal (or the terminal specified in a network-operator logon for TOLTEP). Requested information includes the name of the control terminal (unless a logon was used to invoke TOLTEP, in which case the terminal specified in the logon is used as the control terminal), the name of the alternate printer (optional), the name or names of the nodes to be tested, and the tests to be executed. If TOLTEP is invoked by any means other than the MODIFY command, the system console operator is notified of the terminals and other nodes specified in the request to TOLTEP and is asked for permission for the testing to proceed. Any additional nodes identified by the control terminal for testing must also be approved by the system console operator. (The user-written authorization exit routine must permit TOLTEP to connect and disconnect terminals involved in tests.)

Once authorization has been given by the system console operator, TOLTEP connects the appropriate terminals and begins testing the specified nodes according to instructions from the control terminal and the appropriate OLTs.

For information about how to run TOLTEP, see *DOS/VS and OS/VS TOLTEP for VTAM*.

### ***Reliability and Availability Support***

The purpose of VTAM's reliability and availability support is to maintain the operation of the telecommunication network. This support attempts to prevent problems, and if that is not possible, to minimize the impact of the problems. VTAM's reliability and availability support includes:

**Error Detection and Feedback:** Before attempting to act upon any request, VTAM analyzes it for any erroneous information. If an error is detected, VTAM returns the request along with an indication of the error.

**NCP Initial Test:** VTAM allows a communications controller to be tested before an NCP is loaded. This test can be used to identify problems in a controller before it is made an active part of the system. This testing is optional for local communications controllers; it is required for remote communications controllers.

**Storage Management:** VTAM controls the allocation of much of the storage required for its operation. Using this control, VTAM permits the queuing of requests and attempts to avoid storage interlocking conditions. (A storage interlocking condition is one in which so much of VTAM storage has been devoted to the initiation of request processing that not enough is available to complete that processing.)

**Hardware Error Recovery:** Using the facilities of the operating system and of the NCP, VTAM attempts to recover from some errors. If recovery is not possible, VTAM records a permanent error and attempts to reallocate resources to reduce the impact of the failure. If recovery is possible, processing continues, but a count is maintained of the temporary error.

**Software Error Recovery:** VTAM tries to recover from some errors. If recovery is not possible, VTAM first attempts to isolate the problem and continue processing. If VTAM cannot continue processing, it attempts an orderly shutdown of the telecommunication system so as not to affect the non-telecommunication jobs in the host system.

**NCP Slowdown:** VTAM recognizes NCP slowdown and assists the NCP to return to normal operations.

**Configuration Restart in OS/VS and DOS/VS Release 33:** If an error occurs in the communication system, VTAM attempts to restore the system to its prefailure status. If it cannot, VTAM provides facilities that allow the user to restore the network to either its initial or prefailure status. VTAM also allows manual switched backup for the CPU and communications controllers.

**Configuration Restart in DOS/VS Release 32:** If an error in the communication system causes an NCP to terminate, VTAM attempts to restart the affected NCP.

These operations are discussed in more detail below.

## **Error Detection and Feedback**

All requests from the network operator and from application programs are tested for errors. If an error is detected, the request and an indication of the error are returned to the requester. If an invalid condition is encountered during processing of the request, VTAM stops processing the request and returns an error indication to the requester. No further processing is done for the request.

If the requester is the network operator, VTAM first checks the syntax of the command. If there is an error in syntax, the command is rejected with a message to the network operator; the message describes the error and specifies where it occurred. If the syntax is valid, the requested function is validated. (For example, a request to activate a minor node whose major node is inactive is an invalid request.) If the function requested is invalid, VTAM sends an error message to the operator.

If both the syntax and the function are validated but VTAM encounters an error while processing an operator command, the network operator is notified of the unsuccessful operation and of the reason for the failure.

In addition to notifying the network operator of error conditions directly attributable to operator commands, VTAM transmits messages to the console describing any unusual or error conditions that affect the operation of the telecommunication system, such as permanent hardware errors and the abnormal termination by VTAM of an application program.

If an error condition is found in either an application program's request or in the processing of such a request, VTAM attempts to notify the application program of the error. Notification is made by scheduling a user-written exit routine or by a return code. VTAM has numerous return codes to help isolate the reason for the error. VTAM avoids abnormally terminating application programs if possible. Using the exit routines and the return codes, the application program can attempt to correct the error, bypass the error, or terminate processing.

## **NCP Initial Test**

The NCP initial test is a facility of the communications controller. It is exercised automatically for remote communications controllers; it is an option for local communications controllers. If testing is specified in the generation of an NCP for a local communications controller, VTAM automatically invokes the test program prior to loading that NCP. If testing is not specified, initial testing of the NCP is not performed for a local communications controller.

## **Storage Management**

VTAM has a number of storage pools as noted in "VTAM Buffering" in Chapter 3. The user can specify the size of each pool and a threshold value for each.

These storage pools are used to obtain buffers to transfer data between the host computer and the network and to obtain storage for VTAM control blocks needed to service each telecommunication request. Because these pools are in VTAM storage and are managed

by VTAM, each application program need not be significantly concerned with providing storage in its own partition or address space for VTAM. (The application program must provide storage only for the work areas and VTAM control blocks, such as the ACB, EXLST, NIB, and RPL, that it uses directly.)

VTAM manages its storage pools by determining when storage should be requested, the amount of storage to request, and whether sufficient space is available to meet the request. VTAM also enables requests for buffers to be queued if sufficient space is not available to meet demands. By queuing storage requests, VTAM reduces the need to abnormally terminate an application program when insufficient storage is available at a particular time for its operation.

The pool used to satisfy a storage request within VTAM depends upon the type of request. For example, the pool used to obtain storage for data buffers is not the same as the one used for control blocks, and the pool used to obtain storage for a control block built during the initial stages of processing a request is not the pool used to obtain storage for a control block built during the completion of that request processing. By discriminating among storage requests, VTAM attempts to avoid storage interlocking conditions. If one pool should become temporarily exhausted, the VTAM procedures that need storage from that pool may be forced to wait until it is replenished, but procedures that use storage from other pools can continue processing.

In OS/VS, VTAM also allows the user to specify a maximum threshold value for each pool. When a pool is operating above its threshold, only priority requests (determined by VTAM) for storage from that pool are satisfied. Use of the threshold option can assist VTAM to maintain the availability of the telecommunication system by allowing critical telecommunication functions to continue even if the noncritical functions must wait temporarily. When storage is not available in a pool, VTAM queues storage requests for that pool; then, when storage is available, those requests are serviced.

Establishing buffer limits for terminal and application-program connections is another option that affects VTAM's storage management. Using this option, the user can prohibit low-priority jobs from monopolizing VTAM's storage pools.

In addition to managing storage by controlling its allocation, VTAM attempts to protect its storage from unauthorized access. VTAM components, control blocks (except for application programs' ACBs, NIBs, and RPLs), and buffers reside in storage protected from nonprivileged users by an operating system key.

## **Error Recovery**

VTAM attempts to recover from both hardware and software errors. Both kinds of recovery are discussed below.

**Hardware ERPs:** When an I/O error interruption occurs, the appropriate error recovery procedures (ERPs) are invoked (within the operating system or VTAM for errors on local devices and within the NCP for errors on remote devices). The ERPs attempt to determine the type of error and to recover from it.

When a permanent error is encountered for a local device, the ERP notifies the network operator of the problem and creates an error record as described in "Serviceability Aids," earlier in this chapter. When a temporary error is encountered, no message is sent to the network operator, although a count is maintained of the temporary errors for that device.

Each NCP provides ERPs for the devices attached to it. When an NCP has completed error-recovery processing, it transmits an error record to VTAM for recording on the operating system's error-record data set. For a permanent error, a message describing the problem is sent to the network operator.

**Processing Software Errors:** When VTAM encounters a software error, it attempts to determine whether the error resulted from action on the part of the user, the operating system, or VTAM itself. If it is a user error, it is handled as explained in "Error Detection and Feedback," earlier in this chapter.

If VTAM cannot determine whether the problem is a user error, the access method attempts to recover from the error. If recovery is not possible, VTAM attempts to isolate the cause of the problem and to bypass it. If this partial recovery is not possible, VTAM abnormally terminates, attempting to close down the telecommunication system without impacting the rest of the operating system. Note that if a problem can only be alleviated by abnormally terminating the application program associated with the problem, VTAM terminates the application program. Terminating the application program is an attempt to reduce the impact of an error on the total telecommunication system; terminating one application program still permits other applications to continue to use the telecommunication system.

### NCP Slowdown

If telecommunication activities overload an NCP and exhaust its buffers, the NCP automatically enters slowdown mode. In slowdown mode, the NCP reduces its acceptance of input data (from both the network and the host computer) and attempts to increase the rate of output. VTAM recognizes an NCP slowdown, and, to facilitate NCP recovery, VTAM stops scheduling input and output requests for that NCP, although VTAM does continue to read from the NCP. During this slowdown processing, VTAM continues to accept requests from application programs, but it queues those requests that are directed to nodes in the network controlled by the NCP in slowdown mode. When the NCP has recovered and is no longer in slowdown mode, the queued requests are processed and sent to the NCP.

### Configuration Restart in OS/VS and DOS/VS Release 33

In OS/VS and DOS/VS Release 33, VTAM provides the capability to restore the VTAM network after a failure occurs. The following forms of network recovery are provided:

*Immediate Configuration Restart* is an attempt by VTAM to restore a local or remote NCP after a failure occurs that requires reloading the NCP, or to restore a physical unit and its associated logical units when a loss of contact with VTAM occurs.

*Delayed Configuration Restart* is the facility to restore the network to either its initial status or prefailure status after a VTAM failure, host operating system or host computer failure, or an NCP failure from which VTAM was unable to recover, or entry of a HALT or VARY NET,INACT command by the network operator.

*Manual Switched CPU and Communications Controller Backup* is a facility that allows a user to switch to an alternate CPU or communications controller if an error occurs.

**Immediate Restart of an NCP:** VTAM maintains a record of the configuration of the VTAM network. When an NCP fails and the error is not a channel error, the NCP is dumped if the PCCU definition statement specifies AUTODMP=YES or if the network operator requests the dump. The NCP is then reloaded if AUTOIPL=YES is specified or if the operator requests that the NCP be reloaded. If the NCP is not to be reloaded, it is deactivated.

After the NCP is reloaded, VTAM reinstates the NCP's network as it was before the failure. VTAM application programs are notified of the state of the network as it affects their communication. The network operator is also informed of the status of the network during restart.

**Immediate Restart of Physical Units:** When an error occurs in a physical unit that causes loss of contact with VTAM, or when an NCP to which the physical unit is attached fails and is restarted, VTAM attempts to reestablish contact with the physical unit and logical units that were active at the time of the failure. If the restart attempt fails, the physical unit is deactivated.

VTAM application programs that are in session with logical units associated with a failing physical unit are notified (by their LOSTERM exit routines) of the failure. If the restart attempt is successful, they are again notified by their LOSTERM exit routines that the session has been terminated but can be reestablished. If the restart fails, they are notified that the session is terminated and cannot be reestablished. The network operator is informed of the status of the physical unit during restart.

**Delayed Configuration Restart:** When the network is defined to VTAM, the user can define a VSAM data set for any major node in the network in which VTAM records the status of each associated minor node. When a major node is deactivated, the data set is kept for use when the major node is reactivated. In addition, when VTAM is started, the network operator can specify a VSAM data set in which VTAM maintains a list of active major nodes. When VTAM is stopped, these data sets are kept for use when VTAM is restarted. These types of VSAM data sets are called *configuration restart data sets*.

After normal deactivation or a failure that cannot be handled by immediate configuration restart (a VTAM, host computer, or host operating system failure, or an NCP failure from which VTAM could not recover), the network operator can restore the VTAM network to either its initial status (as defined during network definition) or its prefailure or predeactivation status (as defined by the configuration restart data sets). To restore the network to its initial status, the network operator specifies COLD and a member of the VTAM definition library on the START command or COLD on the VARY commands used to reactivate the major nodes in the network. To restore the network to its prefailure or predeactivation status, the network operator specifies WARM with the list of previously active major nodes.

**Manual Switched CPU and Communications Controller Backup:** Use of configuration restart data sets makes it possible for a user, after a host computer or host system failure, to transfer the data sets (and major node definition data sets if not already duplicated) to an alternate CPU, add these data sets to the VSAM catalog, manually switch the desired portion of the network to the alternate CPU, and restore the network configuration to its prefailure status. Similarly, a user can use the configuration restart data sets to restore the network configuration after manually switching to a backup communications controller.

**Configuration Restart in  
DOS/VS Release 32**

In DOS/VS Release 32, VTAM provides the capability to restart an NCP in a communications controller after a failure occurs in that controller. An NCP can be restarted only if the original failure was in the communications controller. If the failure was in the host computer, configuration restart cannot be used.

When an error occurs in a communications controller, VTAM's ERPs determine whether the channel (for a local attachment) or the line (for a remote attachment) failed. If neither failed, VTAM initiates the NCP dump program (if automatic dumping was specified in the generation of the NCP or if requested by the network operator). If the network operator specifies the NCP is to be reloaded, VTAM attempts to reload and restart it. NCP initial testing is performed if the communications controller is remotely attached or if the communications controller is locally attached and the NCP was generated with the test option requested. After the NCP is restarted, VTAM attempts to reinstate the network status at the time of the failure. That is, nodes active at the time of failure are reactivated (if possible), any remotely attached communications controllers

active at the time of the failure are restarted if possible, any line-scheduling parameters modified by the network operator are reestablished, and (if the NCP included PEP) lines are reinstated to the mode they were in when the failure occurred.

If an NCP is successfully restarted, VTAM attempts to restore the connections between application programs and terminals:

- For application programs that were connected to terminals in basic mode, the connections are maintained by VTAM for terminals that were successfully reactivated. Outstanding I/O requests for these terminals are purged, and the application programs must resubmit their requests.
- For application programs that were connected to terminals in record mode, the application programs must issue CLSDST and OPNDST macro instructions (in that order) for the terminals.
- For application programs connected to terminals that could not be reactivated, the connections are lost; the application programs should issue a CLSDST macro instruction for those terminals but can continue using unaffected connections.





## CHAPTER 7. VTAM PLANNING CONSIDERATIONS AND REQUIREMENTS

The first part of the chapter states the machine and program requirements (including NCP requirements) for using VTAM. The second part of the chapter discusses ways a user can plan for and take advantage of features and facilities in VTAM.

### Machine Requirements

This section discusses the machine requirements for a telecommunications system with VTAM. It specifies which units can be used with VTAM including the central processing units (CPUs) and their required features, communications controller requirements, and terminals and terminal features.

#### *CPU Support*

VTAM is a System/370 access method and runs in a virtual storage environment on one of the following CPUs with the relocation feature:

*Under DOS/VS:* System/370 Models 115, 125, 135, 135-3, 138, 145, 145-3, 148, 155II, 158, and 158 Submodel 2 (available for World Trade countries only).

*Under OS/VS1:* System/370 Models 135, 135-3, 138, 145, 145-3, 148, 155II, 158, 158 Submodel 2 (available for World Trade countries only), 165II, and 168.

*Under OS/VS2 (MVS and SVS):* System/370 Models 145, 145-3, 148, 155II, 158, 158 Submodel 2 (available for World Trade countries only), 158MP, 165II, 168, and 168MP.

The host CPU must be equipped with the Compare and Swap and the Compare Double and Swap instructions. These instructions are available as follows:

Optional on the System/370 Model 135 through the Conditional Swapping Feature 1051.

Optional on the System/370 Model 145 through the Advanced Control Program Support Feature 1001 or the Conditional Swapping Feature 1051.

Standard on the System/370 Models 115, 125, 135-3, 138, 145-3, 148, 155II, 158, 158 Submodel 2 (available for World Trade countries only), 158MP, 165II, 168, and 168MP.

#### *Local 3270s*

VTAM provides communication with IBM 3270 Information Display Systems that are locally attached. The maximum number of 3270s that can be used and any required features are determined by the 3270 and the system, not by VTAM. Appendix A lists the devices and features for local 3270s that can be used with VTAM.

#### *Local 3790s*

VTAM provides communication with IBM 3790 Communication Systems that are locally attached. Appendix A lists devices that may be used in a 3790 system.

#### *Requirements for Communications Controllers*

For remote devices, VTAM requires a local IBM 3704 or 3705 Communications Controller.

VTAM uses only the network control program/VS (NCP) for communications controllers. VTAM uses only the network control mode of the NCP with or without partitioned emulation programming (PEP). VTAM Level 2 requires NCP/VS Version 4, Modification Level 0 or subsequent levels. VTAM Level 2 with enhanced configuration restart requires NCP/VS Version 4, Modification Level 1 or subsequent levels.

VTAM can use both the local and remote communications controllers. A remote communications controller must run in network control mode only and must be attached to a local communications controller on an SDLC line controlled by the network control mode of the NCP in that local controller.

VTAM can be used with the following communications controller models:

IBM 3704 Communications Controller—Models A3 and A4

IBM 3705-I Communications Controller—Models A2, B2, B3, B4, C2, C3, C4, C5, C6, D2, D3, D4, D5, D6, D7, and D8

IBM 3705-II Communications Controller—Models E2, E3, E4, E5, E6, E7, E8, F2, F3, F4, F5, F6, F7, F8, G2, G3, G4, G5, G6, G7, G8, H2, H3, H4, H5, H6, H7, and H8

VTAM does not support the following models:

IBM 3704 Communications Controller—Models A1 and A2

IBM 3705-I Communications Controller—Models A1, B1, C1, and D1

IBM 3705-II Communications Controller—Models E1, F1, G1, and H1

Except for storage capacity, the features required on a communications controller do not depend upon VTAM. Required features depend upon the intended application of the communications controller (including local or remote attachment) and the type of control program to be used (NCP with or without PEP).

**Channel Adapter Support:** Support of channel adapters depends on the communications controller and the NCP.

The local 3704 is equipped with the Type 1 Channel Adapter.

The local 3705-I or 3705-II can be equipped with either the Type 1, Type 2, Type 3, or Type 4 Channel Adapters.

**3705 Two-Channel Support:** Using NCP facilities and a Type 3 or Type 4 Channel Adapter, VTAM supports the attachment of two channels to a 3705 in an OS/VS2 tightly coupled multiprocessing environment. This attachment is not available for a loosely coupled multiprocessing environment for two independent CPUs.

VTAM does not support the manual Two-Channel Switch (8002) for the 3705. The user is responsible for support of this feature in a VTAM system.

**Backup (Alternate) SDLC Link between Local and Remote Communications Controllers:** VTAM supports use of an alternate SDLC link to back up the main SDLC link between a local and a remote communications controller. This link can be switched or nonswitched. VTAM notifies the network operator of the main link's failure; the network operator can then activate the backup link. *NCP Generation* describes how to specify a backup link. *VTAM Network Operating Procedures* describes the procedures that the network operator must perform to switch to a backup link.

**Other Features Not Supported:** VTAM does not support speed selection for lines equipped with dual speed modems. It also does not support switched network backup.

### ***Remote Terminals***

VTAM can control remote terminals only if they are attached through a communications controller in network control mode. SDLC, start-stop, and BSC terminals can be attached either to a local communications controller or to a remote communications controller. Appendix A lists all devices and features that can be used as remote terminals by VTAM.

Because VTAM uses the NCP to control and communicate with remote terminals, most requirements, restrictions, or limitations pertaining to device features are those of the NCP, not of VTAM. (See the *NCP Generation* publication for details on device requirements, restrictions, and limitations for NCP.)

## Storage Requirements

Requirements for CPU storage and for disk storage for VTAM data sets can be calculated by using one of these publications, depending on which operating system is used:

*DOS/VS System Generation*

*OS/VS1 Storage Estimates*

*OS/VS2 MVS Overview*

*OS/VS2 System Programming Library: Initialization and Tuning Guide*

*OS/VS2 MVS System Programming Library: VTAM*

Requirements for disk storage for NCP data sets can be calculated by using the *NCP Generation* publication.

Storage requirements for TOLTEP data sets are included as part of VTAM data set storage in the publications mentioned above, since TOLTEP is included in VTAM. Disk storage information for the online tests is provided by the IBM program support representative.

Storage requirements for communications controllers can be calculated by using *IBM 3704 and 3705 Communications Controllers Network Control Program Storage and Performance Estimates (for OS/TCAM, OS/VS TCAM, and OS/VS and DOS/VS VTAM Users)*

## Operating System Requirements

This section discusses VTAM's requirements for each operating system (DOS/VS, OS/VS1, and OS/VS2) in which VTAM can be included.

### *Upward Compatibility*

The VTAM macro language is upward compatible from DOS/VS to OS/VS1 to OS/VS2. In addition, procedures for defining the telecommunication network and network operator commands are similar for all three systems. VTAM's requirements for each of these systems are detailed in the remainder of this chapter.

### *Requirements for DOS/VS*

Generating VTAM in a DOS/VS system requires specification of VTAM as a teleprocessing access method in the SUPVR macro instruction for DOS/VS system generation. VTAM also requires the inclusion of some DOS/VS facilities that would be optional if VTAM were not in the system. Of these facilities required by VTAM, multitasking support must be specified by the user during system generation. The other options are generated automatically if VTAM is specified; they include:

Support for the use of the STXIT macro instructions (all options) by problem programs

Storage-management support for the GETVIS and FREEVIS macro instructions

Use of the PFIIX and PFREE macro instructions for fixing and freeing pages

Inclusion of the relocating loader

Support for the time-of-day clock

All local devices in the VTAM network must be identified during the generation of the operating system. These devices are local 3270s, local 3790s, and the local communications controllers used by VTAM. Communication lines and remote devices need not be defined at system generation if they are to be used only through VTAM. If they are to be used directly by other access methods, they must be defined according to the requirements of those access methods; remote devices so defined are still available to VTAM through the NCP.

A DOS/VS system with VTAM must have at least two partitions: one for VTAM and one for VTAM application programs. Additional partitions may be needed for other VTAM application programs or for programs unrelated to VTAM. The partition containing VTAM must have a priority higher than any other partition that contains a VTAM application program. In addition to DOS/VS tasks required for application programs, three DOS/VS tasks are required for VTAM. A fourth task is required if the network solicitor is used.

VTAM is started under DOS/VS by entering an EXEC command or job control statement for the partition in which VTAM is to run. In addition to the EXEC statement, an ASSIGN command or statement is required if a local communications controller is to be included (that is, if remote terminals are to be included) in the VTAM network. The assignment must make logical unit SYS000 unassigned for the duration of the VTAM job step. Any other commands or job control statements required to start VTAM depend upon factors such as system generation and telecommunication options to be used. See "VTAM Data Sets" later in this chapter for library and file requirements for VTAM under DOS/VS.

### ***Requirements for OS/VS***

To generate VTAM in an OS/VS system requires that VTAM be specified as a teleprocessing access method in the DATAMGT macro instruction for OS/VS system generation.

All local devices in the VTAM network must be identified at system generation. These devices are the local 3270s, local 3790s, and the local communications controllers used by VTAM. Communication lines and remote devices need not be defined at system generation if they are to be used only through VTAM. If they are to be used directly by other access methods, they need to be defined according to the requirements of those access methods; remote devices so defined are still available to VTAM through the NCP. Data set requirements for OS/VS are discussed in "VTAM Data Sets," later in this chapter.

For OS/VS1 Release 5, VTAM runs in a preallocated partition within the pageable supervisor area of virtual storage. For OS/VS1 Release 6, VTAM can run in a user-numbered problem program partition (if fetch protection is desired) or in a system task partition. For OS/VS2 SVS, VTAM runs in a region that is created when the VTAM job step is initiated. For OS/VS2 MVS, VTAM runs as a problem program.

A cataloged procedure must be created for VTAM. This procedure must contain DD statements for SYS1.VTAMLST and SYS1.VTAMLIB. In addition, the procedure must contain statements for any NCP data sets that are to be used by VTAM. It must also contain a DD statement for SYS1.VTAMOBJ and DD statements for the TOLTEP data sets, if TOLTEP is to be invoked. Local 3270s and 3790s, communications controllers, and remote terminals and devices that are part of the VTAM telecommunication network do *not* require DD statements.

Multiple console support (MCS) can be used with VTAM. Any MCS console that is to be used by the VTAM network operator must be assigned a command authority of 2 and a routing code of 8.

It is not recommended that VTAM be run with the Dynamic Support System (DSS). The effects of DSS on VTAM includes DSS time delays that may result in NCPs in the VTAM network automatically closing down the network. Refer to the publication *OS/VS Dynamic Support System* for details on DSS.

If a VTAM application program is to use the OS/VS2 (MVS) authorized path facility, the program must be authorized. See *OS/VS2 System Programming Library: Supervisor* for how to specify an authorized program.

## Network Control Program Requirements

VTAM uses the network control mode of the network control program/VS (NCP) with the IBM 3704 and 3705 Communications Controllers to control and communicate with remote devices. The NCP is generated with NCP generation macro instructions. These same instructions are then used as definition statements to define the remote fixed part of the network to VTAM. The variable part of the remote network, the switched major nodes, are not described to the NCP; they need only be described to VTAM.

The remainder of this chapter contains details on VTAM's use of the NCP. Refer to the publication *Introduction to the IBM 3704 and 3705 Communications Controller* for a description of the NCP and its facilities. Refer to the *NCP Generation* publication for details on NCP generation macro instructions and NCP generation procedures.

### ***NCP Functions Required by VTAM***

Several types of NCPs may be needed in a VTAM telecommunication system:

- An NCP for a local communications controller to which no remote communications controller is attached
- An NCP for a local communications controller to which a remote communications controller is attached
- An NCP for a remote communications controller
- An NCP with partitioned emulation programming (PEP)

Whatever the NCP or NCPs used, some functions that are optional for the NCP are required by VTAM. These functions are part of the NCP's dynamic control facilities that are created during NCP generation, and they are specified in the NCP's SYSCNTRL statement. The dynamic control facilities that must be specified for start-stop or BSC lines or certain types of operator control include facilities to:

- Change mode settings for a device
- Disconnect a dial connection
- Request a device to yield control of a line
- Reset a device if data transfer has not taken place
- Reset a device if data transfer has taken place
- Reset a device immediately

## *Defining the NCP to VTAM*

To define an NCP and its remote network to VTAM, the following information must be specified in the definition (also the generation) statements:

- The capabilities of the NCP
- The interface between the NCP and VTAM
- The network configuration

NCP generation macro instructions and the PCCU definition statement supply this information to VTAM. Two additional VTAM-only definition statements, VIDLIST and VTERM, apply only to BSC and start-stop terminals. They are described in Chapter 8.

The PCCU statement, used only by VTAM, identifies the communications controller into which the NCP would normally be loaded when activated. One PCCU statement is required for each NCP defined to VTAM.

This statement also defines the functions VTAM is to provide for the NCP. These functions can include:

- Dumping the communications controller automatically following an unrecoverable error in that controller. If a dump is to be taken, a pointer to the name of the data set to contain that dump is specified in the PCCU statement.
- Reloading and restarting the NCP automatically following an unrecoverable error in the communications controller.
- Invoking the NCP initial test routine automatically prior to loading the NCP. (Initial testing is optional only for NCPs for local communications controllers; initial testing is performed automatically whenever a remote communications controller is loaded.)

If the NCP is for a remote communications controller, this statement also provides a link to the NCP for the local communications controller. A PCCU statement for an NCP for a remote controller specifies the name of a PU or INNODE statement in another NCP deck. This deck defines the NCP for the local communications controller, and the INNODE or PU statement defines the remote communications controller.

In addition to using the PCCU definition statement, VTAM requires information contained in the NCP generation macro instructions (also called VTAM definition statements). Most information required by VTAM is also required by the NCP, although some additional data is required by VTAM alone.

## *Pacing*

VTAM enables the user to control the rate of data flow through the network path joining a VTAM application program and a logical unit. This control is called pacing. Pacing is intended to protect a node that is receiving data from being overloaded by too much input. Using the specifications supplied in an LU statement by the user, both VTAM and the NCP can limit the number of messages transmitted to a particular logical unit before a response is returned that the messages have been processed. For each logical unit, the user can specify pacing requirements for the transmission of messages from VTAM to the NCP and for the transmission of messages from the NCP to the logical unit.

## *Support for SNA Switched Networks*

This section describes VTAM support for dial-in and dial-out operations in a switched network using SDLC lines. (For information on switched network support for start-stop and BSC terminals, see Chapter 8). This section discusses:

- Defining a switched major node to VTAM
- Generating a group of switched lines in the NCP for dial-in and dial-out operations

Verifying node IDs

Performing dial-in operations

Disconnecting physical units and logical units with or without automatic call termination

### **Defining a Switched Major Node**

A switched major node is a logical grouping of physical units and logical units that are serviced by SDLC switched line groups. Each switched major node must be defined to VTAM by these statements:

The VBUILD statement defines the start and limitations of each major node.

The PU statement describes a particular physical unit in the switched major node and its disconnection procedure (where it has automatic call termination).

The PATH statement (for dial-out operation only) describes such information as a telephone number to be used, a group name (for a group of switched lines generated in the NCP), and a redial count for devices with automatic dialing.

The LU statement describes characteristics of each logical unit associated with the physical unit.

Each physical unit and its associated logical units defined to VTAM in a switched major node can perform dial-in operations, but only physical units with a PATH statement can perform dial-out operations. (A PATH statement is also required for physical units that may be dialed out from the host and dialed in to the host.)

### **Generating Switched Line Groups in the NCP**

Switched line groups are defined as the last part of the NCP generation. Each switched line group is used to service physical units and their associated logical units in switched major node dial-in and dial-out operations. The following macro instructions must be used to define a switched line group:

The GROUP macro instruction defines the start of a group of switched lines, the line discipline (for example, SDLC), and whether the lines have dial support.

The LINE macro instruction describes whether a line has automatic or manual dial support and whether the line has dial-in, dial-out, or dial-in/out capability.

The PU macro instruction describes the limitations of each line with respect to what type of physical unit can be supported on the line.

The LUPOOL macro instruction states the total number of logical units that can be supported by the group of switched lines being defined.

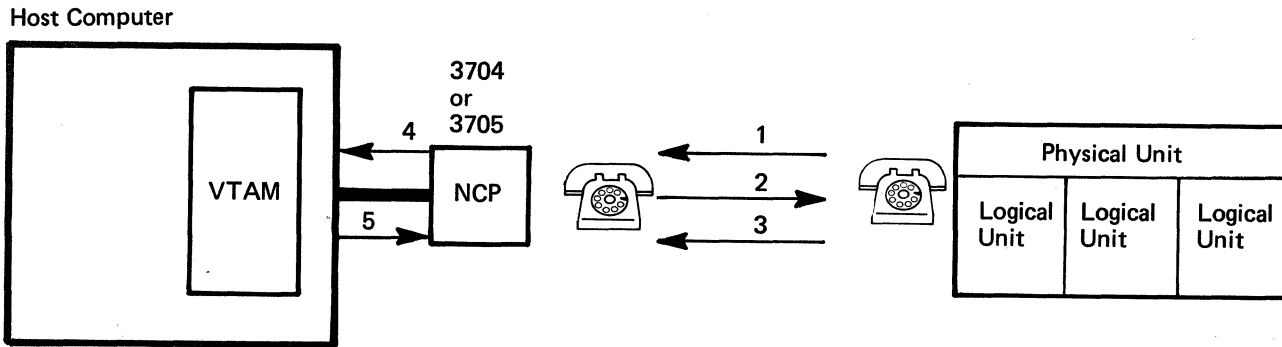
### **ID Verification for Switched SNA Major Nodes**

During dialing operations, the IDs of the physical unit and the host computer are checked. If each ID is valid, normal connection procedures occur to establish connection between a logical unit and an application program. (See Figure 7-1.)

In a dial-in operation, when a dial port is found to handle the dial-in request, the NCP sends an XID (exchange ID) command to the physical unit that requests connection. The physical unit responds with its ID to be checked. The physical unit's ID is then sent to VTAM which validates the ID. Finding that the physical unit's ID is valid, VTAM sends an Activate Physical Unit command containing the host ID to the physical unit.

The dial-out operation verifies the ID in a similar manner. After a dial line is established from a dial port to the physical unit, the same series of events for checking IDs occurs.





1. A telephone is used to dial a dial port of a communications controller sending a signal to the NCP (if the dial port does not answer, other dial ports are called.)
2. The NCP sends an XID (exchange ID) to the physical unit.
3. The physical unit responds with its identifier to be checked.
4. The NCP sends the physical unit's identifier to VTAM to be checked.
5. VTAM checks the identifier and returns information pertaining to the physical unit. The connection procedure from this point on is the same as that for a physical unit on a nonswitched line.

Figure 7-1. Dial-In Operation

### Dial-In Operation

A dial-in operation can be performed by any physical unit defined in a switched major node (see Figure 7-1). To initiate a dial-in operation:

The dial port of the 3704 or 3705 that is being called must be active.

The switched major node containing the physical unit and logical unit must be active.

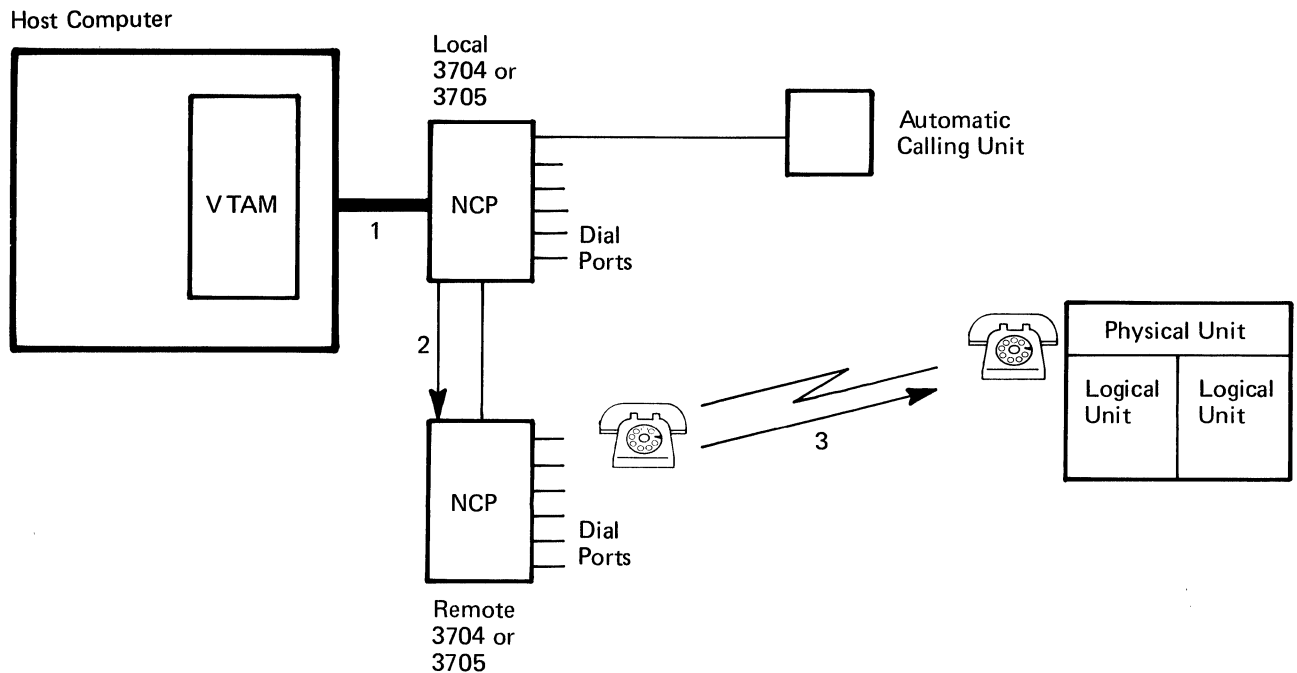
The VTAM application program to which connection is requested must be active with a LOGON exit routine or a pending OPNDST with OPTCD=ACCEPT to handle the dial-in request from the logical unit.

The dial-in operation can be accomplished by manually dialing a dial port of the communications controller or by using a program of a subsystem controller (for example, a 3791 controller can be programmed to dial ports) to dial a port of a local or remote communications controller. As soon as a port is found to handle the dial-in operation, host computer and physical unit IDs are exchanged. The rest of the connection procedure is the same as for physical units and logical units on nonswitched lines.

As with logical units on nonswitched lines, following the dial-in operation, the logical unit on a switched line can become connected to a VTAM application program as the result of: (1) a logon from the logical unit, or (2) a previous network operator-initiated logon, or (3) a VTAM-initiated (automatic) logon. In the case of a network operator- or VTAM-initiated logon for a logical unit that must dial in, VTAM holds the request until the logical unit dials in. An application program-initiated request for connection to a dial-in logical unit is unsuccessful unless the associated physical unit had dialed in before the request was issued.

### Dial-Out Operation

A dial-out operation is performed by VTAM on behalf of an application program to connect it with a specific logical unit (see Figure 7-2). The conditions for performing the dial-out operation are similar to the dial-in operation with the exception that the application program must issue an OPNDST. Dial-out operations can be performed only on those physical units and logical units that have PATH statements associated with them in the VTAM definition.



- 1 and 2. VTAM sends a message (containing a telephone number, a line to use, and, if an automatic calling unit is attached, a redial count) to a communications controller to dial-out to a particular physical unit.
3. The communications controller automatically or the operator manually dials the telephone number of the physical unit and starts to verify the identification of the physical unit.

Figure 7-2. A Path For a Dial-Out Operation

To accomplish a dial-out, VTAM uses the path table (built from the PATH statement during VTAM definition) and the switched line groups. When a request is made to dial out to a particular physical unit, VTAM:

- Picks a usable line in the switched line group and specifies the line, a telephone number, and for automatic dialing, a redial count.

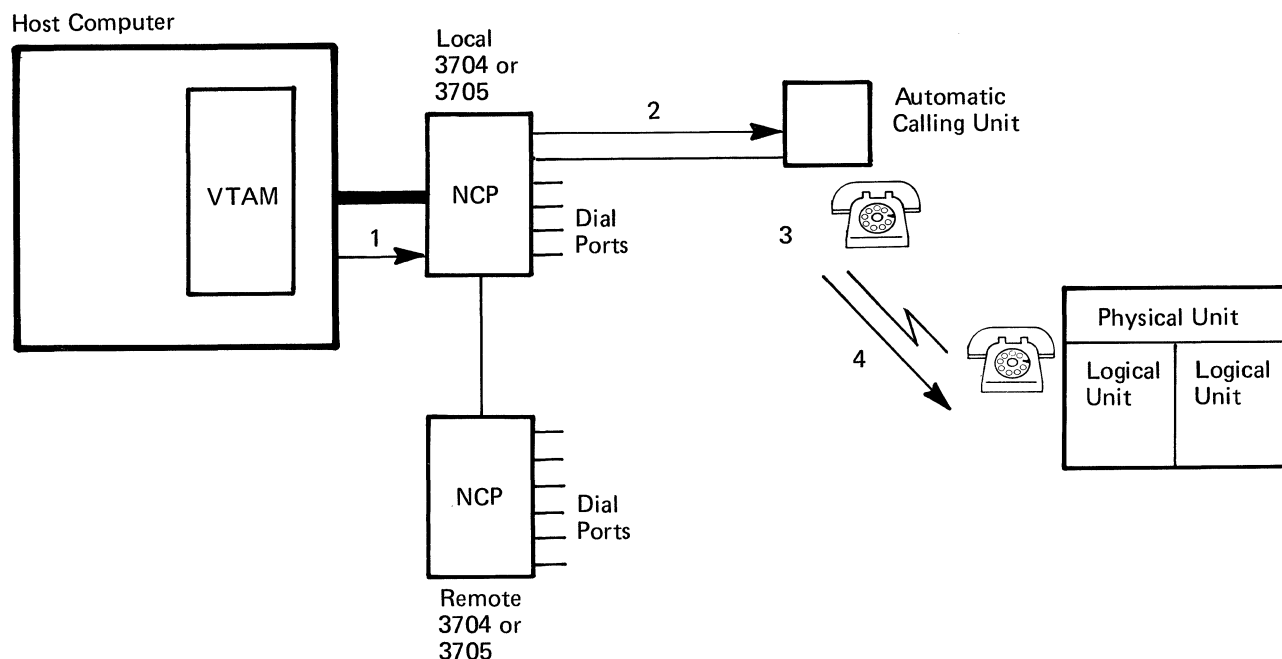
- Finds another usable line in the path (until connection is made or all the lines are used in the path) if the previous line cannot be used.

- Uses another path if the previous path fails (see Figure 7-3).

- Rejects the OPNDST with OPTCD=ACQUIRE if all paths are unusable. The VTAM application program can determine the reason for the failure and try again later.

Paths can be controlled by operator commands. By coding the path identifier (PID) or a group identifier (GID), a path for a particular physical unit or a group of paths for a major node can be activated or deactivated. Group identifiers are defined by the user to classify groups of paths in a switched major node (for example, to separate classes of telephone numbers such as local numbers, WATS lines, long distance, and direct dial). The path identifier is used to identify each unique path to a particular physical unit.

Another consideration in a dial-out operation is whether the dialing operation is manual or automatic. When manually dialing, a message is sent to the console of the network operator telling him to dial a telephone number, using a specific line. When an automatic calling unit is attached to a communications controller, the unit uses the line given and dials the telephone number a specified number of times.



1. VTAM sends a message to a communications controller to perform a dial-out to a specific physical unit.
2. A signal containing a telephone number is sent to an attached automatic calling unit.
3. The automatic calling unit dials the telephone number of the physical unit.
4. A signal is sent to the physical unit that will start the connection procedure.

Figure 7-3. An Alternate Path For a Dial-Out Operation with an Automatic Calling Unit

#### Disconnecting Physical Units and Logical Units in a Switched Major Node

The user may specify that all physical units and logical units in a switched major node have automatic call termination. When all sessions between logical units and application programs are terminated, the physical unit is disconnected.

Disconnection procedures can be changed for each physical unit in the switched major node using the DISCNT operand on the PU statement during VTAM definition. If DISCNT=YES is specified, automatic call termination is used as described above. If DISCNT=NO is specified, the physical unit remains connected to VTAM unless all logoffs from associated logical units specify that the physical unit is to be disconnected when the last logical unit is disconnected. See "Holding the Physical Unit Connection" in Chapter 3 for a description of specifying disconnection of the physical unit on a disconnection request.

#### Partitioned Emulation Programming (PEP)

VTAM can use the NCP with partitioned emulation programming (PEP). When PEP is in use in the NCP, VTAM does not provide services for the network serviced by emulation mode. VTAM does provide three facilities that affect the emulation mode of an NCP with PEP: dumping the communications controller, reloading the controller, and changing line assignments.

If the network operator invokes the NCP dump utility program (using VTAM's MODIFY command), the dump disrupts the emulation function as well as the network control function when an NCP with PEP resides in the communications controller. After a dump is taken the NCP has to be reloaded before it can be used. (An NCP with PEP is loaded automatically by VTAM when the access method activates an NCP major node defining a network control function that is part of an NCP with PEP.)

VTAM also provides support for the restart capability of the NCP. If an error is encountered in a communications controller and this error causes VTAM to reload the controller, the reloading disrupts emulation because the entire NCP is reloaded, including the emulation function.

VTAM also affects the emulation function in an NCP with PEP by changing line assignments. Using PEP, lines can be assigned to network control mode, emulation mode, or to both. If possible, VTAM assigns a line to both modes. VTAM's support for such lines is as follows:

If the line is not active for VTAM, it is automatically assigned to emulation mode.

Line assignments are changed in response to activation and deactivation requests from the network operator. Lines are assigned to network control mode when they are activated by VTAM, and they are reassigned to emulation mode when they are deactivated by VTAM.

## VTAM Data Sets

VTAM uses a number of data sets (files in DOS/VS) to support a telecommunication system. These data sets contain such items as VTAM modules, VTAM definition statements, NCP modules, and RAS records. Data set requirements vary depending upon the operating system in which VTAM is being used.

### *Data Sets for VTAM under OS/VS*

In an OS/VS system, VTAM uses NCP data sets, operating system data sets, and data sets devoted entirely to VTAM itself. Figure 7-4 is a table of all the data sets used by VTAM in an OS/VS system. The table columns show the following:

**Name:** Specifies the name assigned to the data set.

**VTAM Contents:** Indicates what information in the data set is used by VTAM.

**When Created:** Specifies the latest time that the data set can be created.

**JCL:** Specifies whether a DD statement must be provided for the data set in the VTAM start procedure.

**Comments:** Provides additional information relevant to the data set. If the data set is specified as "required," it is required by VTAM. If it is specified as "optional," it is not required by VTAM.

**Note:** *If an optional data set is not included in a VTAM system, the contents of that data set, and therefore the associated function, are not available to VTAM.*

Refer to the appropriate operating system planning guide for more information on the operating system data sets listed in Figure 7-4. Refer to the publication *Introduction to the IBM 3704 and 3705 Communications Controllers* for additional information on NCP data sets listed in Figure 7-4.

### *Data Requirements for VTAM under DOS/VS*

The table in figure 7-5 shows the file requirements for VTAM under DOS/VS. This table is divided into columns that are defined as follows:

**File Name:** Specifies the name of the file. Except as indicated in Figure 7-5, this name must be specified as shown.

Name	VTAM Contents	When Created	JCL	Comments
<i>VTAM Data Sets</i>				
SYS1.VTAMLIB	Load modules for VTAM; see note 1	System generation	Yes	Only the VTAM load modules are required
SYS1.VTAMLST	VTAM definition statements and start options	Prior to starting VTAM	Yes	Required. See Note 2.
SYS1.VTAMOBJ	VTAM network configuration table	Prior to starting VTAM	Yes	Required. See Note 3.
<i>Operating System Data Sets</i>				
SYS1.DUMP	Records for SVC dumps	IPL	No	Optional
SYS1.LINKLIB	VTAM initialization module, used when VTAM is started	System generation	No	Required
	NCP loader utility program	Prior to starting VTAM	Yes	Required
	NCP dump bootstrap program	Prior to starting VTAM	Yes	Required
	Local communications controller pre-IPL testing modules	NCP generation	Yes	Required if the controller is to be tested automatically prior to the loading of the NCP by VTAM.
SYS1.LOGREC	VTAM error records	System generation	No	Required
SYS1.LPALIB	VTAM load modules, user-written authorization and accounting exit routines, and logon mode and interpret tables to be loaded into the shared link pack area	System generation	No	Required. An OS/VS2 data set only.
SYS1.MACLIB	VTAM macro definitions	System generation	No	Required
SYS1.NUCLEUS	VTAM resident SVCs and abnormal termination modules	System generation	No	Required
SYS1.PARMLIB	VTAM space parameter for IPL	System generation	No	Required.
SYS1.PROCLIB	VTAM cataloged procedure	System generation	No	Required
SYS1.SVCLIB	VTAM non-resident SVCs and ERPs for local devices	System generation	No	Required. An OS/VS1 data set only.
SYS1.TRACE	GTF trace records for VTAM	System generation	No	Optional

**Notes:**

1. SYS1.VTAMLIB is referred to as the *VTAM load module library*. For OS/VS1, SYS1.VTAMLIB contains VTAM load modules for the VTAM virtual partition; for OS/VS2, it contains VTAM load modules for the VTAM private address space. In OS/VS1, this library can contain the interpret table, or tables, containing logon descriptions and any installation-coded logon-routines specified in these tables; logon mode tables; and authorization and accounting exit-routines. In OS/VS1 and OS/VS2, this library can contain the network solicitor.
2. SYS1.VTAMLST is referred to as the *VTAM definition library*. Definition statements for each major node are filed in this library. Each major node is a separate member of SYS1.VTAMLST.  
Start options are filed in members under the name of either ATCSTRxx or ATCCONxx (where xx are two-digit numbers specified by the installation). ATCSTRxx members contain lists of major nodes to be activated automatically when VTAM is started. See "Defining the Network to VTAM", in Chapter 3, for details on VTAM definition statements and VTAM start options.
3. When a major node is activated, VTAM builds a table describing the node from the information supplied by definition statements. When a major node is deactivated, its table is deleted by VTAM. These tables are used to maintain the current status of all minor nodes in the telecommunication system.  
To reduce the time needed to construct these tables, VTAM stores a copy of each table on SYS1.VTAMOBJ the first time each major node is activated. This copy is then used whenever the major node is again activated.

*Note:* A major node can be modified merely by modifying its definition statements and refiling them under the same member name on SYS1.VTAMLST. If a member is refiled, the copy of the corresponding table on SYS1.VTAMOBJ must be deleted (using an operating-systems utility program that can delete a member of a BPAM data set). If the copy is deleted, the next time the major node is activated, VTAM builds a new table (based on the modified definition) and stores a new copy on SYS1.VTAMOBJ.

Figure 7-4. (Part 1 of 2). Table of Data Sets Used by VTAM under OS/VS

Name	VTAM Contents	When Created	JCL	Comments
<i>NCP Data Sets</i>				
NCP dump	Dump records for the NCP	When VTAM is started	Yes	Required if VTAM is requested to provide a dump of an NCP. One such data set must be provided for each NCP that is to be dumped simultaneously.
NCP load library	NCP load modules	NCP generation	Yes	One such data set must be created for each NCP used by VTAM. See note 4
Configuration restart data sets	VTAM status of network nodes	Prior to starting VTAM	Yes	Required if a warm restart is to be used
INITEST	Local communications controller pre-IPL testing modules	NCP generation	Yes	Required if the controller is to be tested automatically prior to the loading of the NCP by VTAM
<i>TOLTEP Data Sets</i>				
TOLTEP CDS	Configuration data sets	Prior to invoking TOLTEP	Yes	Required if TOLTEP is to be executed
TOLTEP OLTs	Online terminal tests for TOLTEP	Prior to invoking TOLTEP	Yes	Required if TOLTEP is to be executed

**Notes:**

4. These data sets are referred to as the *NCP load module libraries*.

Figure 7-4 (Part 2 of 2). Table of Data Sets Used by VTAM Under OS/VS

**File Identification:** Specifies the identification of the file on the volume.

**Contents:** Indicates what VTAM uses in this file.

**Comments:** Provides additional information relevant to the file. If the file is specified as "required," it is required by VTAM.

Refer to the publication *Introduction to DOS/VS* for additional information on the DOS/VS logical units and files. Refer to the publication *Introduction to the IBM 3704 and 3705 Communications Controllers* for additional information for NCP data requirements.

VTAM also has data requirements pertaining to DOS/VS libraries. Whether these libraries are system or private libraries depends upon system generation options selected by the installation. Library requirements for VTAM under DOS/VS are described in Figure 7-6.

File Name	File Identification	Content	Comments
DIAGFILE	INITST	Communications controller pre-IPL testing modules	Required if the controller is to be tested automatically prior to the loading of the NCP by VTAM
NCP load file	Note 1	NCP load modules	One such file must be created for each NCP used by VTAM. See note 2
NCPDUMP	Note 3	Dump records for the NCP	Required if VTAM is requested to provide a dump of an NCP. One such file must be provided for each NCP that is to be dumped simultaneously
TRFILE	TRFILE	VTAM trace records	Required. Must be assigned to SYS001

*Notes:*

1. File names and file identifications are created during NCP generation.
2. These files are referred to as the *NCP load libraries* in this publication.
3. File identifications are created during NCP generation.

Figure 7-5. Table of Files Used by VTAM Under DOS/VS

## Sharing Telecommunication Resources

VTAM permits resources in its telecommunication system to be shared. The degree of sharing depends upon such factors as the resources themselves and the user's management of these resources.

### *Resources That Can Be Shared*

Using VTAM facilities, application programs can share terminal components, terminals, cluster control units, communications controllers, NCP facilities, communication lines, and data channels. Of these elements, terminals (or components) and application programs can be thought of as end nodes, while the remainder of the elements are part of the path connecting the end nodes.

Sharing of end nodes is different from the sharing of path elements. End nodes are shared for each connection; path elements can be shared simultaneously. That is, an active terminal (an end node) can be connected to or queued for logon to only one application program at a time. If an active terminal is not connected or queued for logon to an application program, it is available for connection to any application program.

On the other hand, VTAM does not explicitly connect path elements to end nodes. Instead, VTAM employs path elements on behalf of end nodes only as long as needed to complete a specific I/O request. For example, more than one terminal can be attached to the same multipoint line, and each terminal can be connected to, or queued for logon to, a different application program. Thus, the line (path element) is used simultaneously by several application programs.

DOS/VS Library	Content	Comments
Core image library	VTAM modules; NCP tables and block handler routines; configuration data sets and online terminal tests for TOLTEP	Required. See Note 1.
Configuration restart data sets	VTAM status of network nodes	Required if a warm restart is to be used.
Procedure library	VTAM start procedure	Optional.
Source statement library	VTAM definition statements and start options	Required. See Note 2.

**Notes:**

1. In this publication, *VTAM load module library* refers to the VTAM contents of this library. This library also contains the interpret table, or tables, containing logon descriptions and any installation-coded logon-routines specified in these tables; authorization and accounting exit routines; and the network solicitor.
2. In this publication *VTAM definition library* refers to the VTAM contents of this library. Definition statements for each major node are filed in this library. Each major node is a separate book of the source statement library.

Figure 7-6. VTAM's Use of Libraries Under DOS/VS

***Sharing and Managing Resources***

While VTAM allows many resources to be shared, the user can restrict some sharing for such reasons as performance or security. For example, a high-priority terminal might be attached to a nonswitched, point-to-point line to improve access to it, or connection to a security-sensitive application program might be limited to only certain terminals. A user can affect resource sharing when defining the network, generating the NCP, or executing application programs. Resource management for each of these areas is discussed below.

**Managing Resources through VTAM Definition**

The telecommunication network that VTAM controls is the one presented to it through network definition. VTAM provides a number of facilities in network definition that enable a user to affect the sharing of telecommunication resources. These facilities include:

- Logon
- Application program authorization
- Authorization exit routine
- Initial status of nodes

The effects of each facility on sharing are discussed below. Details on these facilities are provided in Chapter 3.

**Sharing through Logon:** Using the terminal-initiated logon capability of VTAM, a user can make all input/output terminals available to all application programs. A network can be defined in which application programs can connect with input and output terminals only in response to a terminal-initiated logon. (Input-only and output-only terminals are available for connection by other means, such as automatic logon or acquisition.)

In practice, some of the terminals might not be available to all application programs. For reasons such as security, some of the terminals might be available only to a restricted set of application programs. Restricting the availability of a terminal can be accomplished by:

- Limiting the number of valid logons for a terminal



Prohibiting some connections through the authorization exit routine

**Application Program Authorization:** Some facilities in VTAM require authorization before they can be used by an application program. These facilities can reduce the availability of resources and should therefore be used judiciously:

*Passing connections:* Passing a terminal to another application program queues the terminal to that application program. As long as the terminal is on the queue, it is not available for communication, either to that application program or to any other application program. Care should be taken to ensure that terminals are on the connection queue no longer than necessary.

*Acquiring terminals:* An application program acquires a terminal by issuing an OPNDST macro instruction with OPTCD=ACQUIRE or issuing a SIMLOGON macro instruction. Care should be taken that only certain application programs can acquire terminals.

*Program operator:* An application program can be authorized to enter VTAM network operator commands (except START and HALT) using the SENDCMD and RCVCMD macro instructions. Such an application program (a program operator) can monitor and control the VTAM network. A user should control which application programs have this capability.

*Block processing for start-stop and BSC terminals:* As compared to message or transmission processing, block processing can result in a greater number of I/O interruptions for the application program. More interruptions, in turn, can result in more delays for path elements such as communication lines and data channels. These delays then decrease the availability of these resources to other application programs.

**Authorization Exit Routine:** Although VTAM allows any terminal to be connected to any application program, a user can code an authorization exit routine restricting specific connections. VTAM passes control to this exit routine during the processing of connection and disconnection requests.

**Initial Status of Nodes:** Using VTAM definition statements and start options, a user specifies the initial status of each minor node in the VTAM system. These minor nodes are defined as initially active or initially inactive. If a minor node is not activated when VTAM is started, the network operator must explicitly activate it before it can be used by VTAM. On the other hand, if a minor node is activated when VTAM is started, network operator intervention is not required, and the node may be made available sooner.

Whatever the initial status specified for a minor node, the minor node cannot be activated until its major node is active. Like minor nodes, major nodes can be activated explicitly by the network operator, or they can be activated automatically (when VTAM is started) without network operator intervention.

## Managing Resources through NCP Generation

Because VTAM uses the NCP to communicate with remote devices, options generated in the NCP may affect VTAM's sharing capabilities. For example, if an NCP session limit specified for a multipoint line is not adequate to service all devices on that line simultaneously, some devices may be unavailable for communications. Refer to the *NCP Generation* manual for more information on NCP options.

## Managing Resources through Application Programs

Once a VTAM system has been generated and defined, the availability of resources can be influenced by the application programs' use of VTAM. An application program can monopolize telecommunication resources that normally would be shared:

As long as a terminal is connected to or queued for connection to an application program, it is unavailable to other application programs.

As long as an application program and a terminal are connected, the number of NCP sessions available for a multipoint line (for remote BSC and start-stop terminals) is reduced by one.

Connection and NCP session, as they pertain to sharing resources, are discussed below.

**Connection and Terminal Sharing:** To achieve maximum use of terminals, the user can ensure that terminals are connected, or queued for logon, only as long as necessary. Using VTAM's OPNDST and CLSDST macro instructions allows an application program to be connected to a terminal only during an input or output operation. This feature is particularly useful for application programs with very long delays between message transmissions. Terminals can be released during the delays for use by other application programs.

**NCP Session and Line Sharing for Start-Stop and BSC Terminals:** While an application program and a terminal are connected in basic mode, it is possible for them to monopolize the communication line. The first I/O request from an application program for a terminal connected in basic mode initiates an NCP session for the terminal. This session is continued until the application program disconnects the terminal. Thus, for most application programs, the duration of an NCP session nearly equals the duration of a connection.

The relationship between connection and the NCP session is particularly significant if the line is multipoint and the session limit (as specified in NCP generation) is less than the number of terminals on that line or if the system is DOS/VS and the line is point-to-point with attached components.

In the case of multipoint lines, it is possible for the number of simultaneous connections to equal the specified NCP-session limit. If they are equal and the session limit is less than the number of attached terminals, communication with a terminal not currently in session is prohibited until a current session on the line is ended by the disconnection of another terminal. If the session limit for a multipoint line equals the number of terminals on that line, contention for the NCP session is reduced.

In the case of components on a point-to-point line under DOS/VS, only one session is permitted at a time for that line.

## VTAM Telecommunication Security

Using VTAM facilities, a user can control the use of specified terminals and application program. Resources to be controlled are identified during VTAM definition; control of the access to these resources can be performed at various points in the telecommunication network and during various stages of telecommunication processing. The user specifies when this control is to be exercised. Depending upon the desired type of control, a user can specify control of sensitive resources in one or more of the following:

The communications controller network control program (NCP)

VTAM

An authorization exit routine

Application programs

The user can control sensitive resources by:

Verifying terminal identifications

Verifying logons

Controlling the connections between application programs and terminals

Controlling which application programs can use VTAM

Restricting an application program's use of VTAM facilities

Protecting sensitive data being transmitted by VTAM

The remainder of this section discusses each control facility in detail.

### ***SNA Terminal Identification Verification on Switched Lines***

The IDs of SNA terminals on switched lines are built from the PU statement. VTAM constructs a 48-bit station ID for a physical unit from the PUTYPE, IDBLK, and IDNUM operands. The SNA terminal ID is of the form:

Bits 0-7: The physical unit type (PUTYPE)

Bits 8-15: Reserved (X'00')

Bits 16-27: A 12-bit binary block number assigned by IBM to the specific device (IDBLK)

Bits 28-47: A 20-bit binary identification number assigned to the particular station being defined (IDNUM)

After a physical unit either logs on, dials in, or answers a dial-out request from an application program, an XID command (exchange ID) is sent to the physical unit. The physical unit responds with its ID which is sent to VTAM for verification. If the ID is valid, VTAM sends its ID to the physical unit by the Activate Physical Unit command and gives the NCP (if the physical unit is a remote terminal) information to support a session (such as physical unit type and segment size used by the physical unit). If the ID is not valid, VTAM disconnects the physical unit. When the identifications have been exchanged, normal connection procedures continue.

### ***Host ID Verification for Physical Units***

VTAM allows a host identification number to be specified when VTAM is started, either in a start procedure or by the network operator. This host identification number or SSCPID can be used by a physical unit when it is initiating connection to the host to determine that it is in touch with an authorized host computer. Use of the host identification number depends on each IBM terminal or terminal system product; see the programming publications for each product to determine use of the host identification number.

### ***BSC and TWX Identification Verification on Switched Lines***

Using the IDLIST and VIDLIST definition statements, the user can identify binary synchronous communications (BSC) or TWX terminals attached to switched lines. Both definition statements are placed in the generation (and definition) deck for the communications controller network control program (NCP) for virtual systems. The IDLIST definition statement indicates those identifications to be verified by the NCP. The VIDLIST statement indicates those identifications to be verified by VTAM in the host computer. (See "Network Control Program Requirements" earlier in this chapter for more information on these definition statements.)

In VTAM under OS/VS, a user can distribute verification authority between an NCP in a communications controller and VTAM in the host computer for dial-in operations. Thus, to conserve NCP resources, a user might specify in an IDLIST statement only those terminal identifications that are used heavily, and let VTAM verify less frequently used terminals. In VTAM under DOS/VS, verification for dial-in operations must be done by VTAM in the host computer and verification for dial-out operations must be done by the NCP. Using IDLIST definition statements, the user can specify the following information for the NCP:

The identification character string for each BSC or TWX terminal on a switched line to be verified by the NCP.

The symbolic name to be assigned to the terminal entering a verified identification character string. A unique name can be specified for each verifiable character string.

The action to be taken if the NCP encounters a character string that is not described in an IDLIST definition statement. The NCP can either pass the unverified identification to VTAM for further verification processing or stop communicating with the terminal.

Using the VIDLIST definition statement, the user can specify the following information for VTAM:

The identification character string for each BSC or TWX terminal on a switched line to be verified by VTAM.

The symbolic name to be assigned to the terminal entering a verified identification character string. A unique name can be specified for each verifiable character string.,

If VTAM is unable to verify a terminal identification, it disconnects the terminal or passes it to the application program designated to handle unidentified terminals. VTAM disconnects the terminal only if it cannot pass the terminal to an application program. An application program is designated to handle unidentified terminals by supplying special operands on a TERMINAL statement in the NCP definition deck. These special operands indicate:

That the TERMINAL statement contains terminal descriptions to be applied to unidentified terminals. (This statement contains the CTERM=YES parameter and the UTERM specification.)

That whenever such a terminal description is applied to a terminal, that terminal should be automatically logged on to a specified application program.

Any further verification of the terminal is then the responsibility of that application program.

To accomplish identification verification for dial-in operations in a VTAM system under DOS/VS, the IDLIST definition statement should not be specified, but the VIDLIST definition statement should be coded. Thus, all identification character strings are passed to VTAM for verification. For dial-out operations, the IDLIST statement (rather than the VIDLIST) is coded since verification is performed by the NCP.

To accomplish identification verification in a VTAM system under OS/VS, three options are available:

Use the method described for VTAM under DOS/VS.

Specify IDLIST definition statements to describe all identifications to be verified by the NCP. Specify in an IDLIST definition statement that all unverified identifications are to be passed to VTAM. Code VIDLIST definition statements to handle the verification of all identifications not verified by the NCP.

Code only IDLIST definition statements, verifying all identifications in the communications controller, and pass no unverified identifications to VTAM.

Verifying all identifications in the host computer relieves the NCP of verification requirements and reduces NCP space requirements. On the other hand, verification in the host increases VTAM's requirements for virtual storage and for CPU time. This option is probably best suited for systems having only a small number of terminals to be verified and expecting only infrequent verification activity. It is also suited for systems whose NCP space and performance requirements are significantly more important than those of the CPU and VTAM.

Dividing the verification responsibility between the NCP and VTAM offers flexibility. The exact division can be weighed against space requirements, processing-time considerations, and verification usage.

Verifying identifications only in the NCP reduces VTAM and CPU requirements, and can free the access method of some processing requirements. This option might be selected for those systems expecting heavy identification verification.

### ***Controlling Connections***

VTAM provides the following facilities to assist a user in controlling connections:

An exit that allows a user-written routine to authorize connection requests during program execution.

A mechanism that allows a user to control which application programs can acquire a terminal.

The ability to tailor logons to user requirements and specifications. This facility includes helping the user to control which terminals and operators can log on to which application programs.

### **Authorization Exit Routine**

The authorization exit routine is executed each time a logon or OPNDST macro instruction requests connection or a CLSDST macro instruction requests disconnection. Upon entry, this exit routine is provided with names of the nodes and the type of request. The exit routine can then determine whether the request is valid. Upon returning to VTAM, the authorization exit routine indicates to VTAM whether the request should be permitted. If the request is valid, it is completed by VTAM; otherwise, it is rejected. See "Authorization, Accounting, and Logon-Interpret Exit Routines" in Chapter 3 for more information on this exit routine.

### **Acquiring and Passing Connections**

An application program can request that a terminal be connected or queued for logon by using one of the following methods:

OPNDST macro instruction with the ACQUIRE option

SIMLOGON macro instruction

CLSDST macro instruction with the PASS option

The user controls these options; that is, only application programs authorized by the user during VTAM definition can issue these macro instructions. Unauthorized application programs can only be connected in response to logons with an OPNDST (with the ACCEPT option) macro instruction. A user authorizes an application program in the APPL definition statement. APPL authorization allows the user to control which application programs can initiate connection requests. The following topic, "Controlling Logons," explains how the user can control connection requests that originate outside an application program.

*Note: These forms of connection can also be controlled through the authorization exit routine. See "Authorization Exit Routine" above.*

## Controlling Logons

VTAM definition procedures and the authorization exit routine allow the user to control which terminals can log on to which application programs. To permit terminal-initiated logons, the user does the following during VTAM definition:

*For the terminal:* Specifies in the GROUP, LINE, CLUSTER, VTERM, TERMINAL, LOCAL, or LU definition statement that a terminal is to be monitored by VTAM for logons and defines valid logons.

*For the application program:* Specifies in the APPL definition statement the name of the application program.

Using the logon information supplied by the user at VTAM definition, VTAM's logon facilities determine which logons should be routed to which application programs. In addition to controlling this determination, the user has two other opportunities to inspect and control the pending connection request. These opportunities are in the authorization exit routine and in the application program to which the logon is directed.

The user-written authorization exit routine is notified of all terminal-initiated logons. Upon entry, this exit routine can make a further determination as to whether the logon is valid. If invalid, the connection request is rejected by VTAM; if valid, the request is passed to the application program. (See "Authorization, Accounting, and Logon-Interpret Exit Routines" in Chapter 3 for a description of the authorization exit routine.)

Finally, having been verified by VTAM's logon facilities and the authorization exit routine, the logon must still be accepted by the application program before a connection is established. The application program can inspect the logon and either accept the request or reject it. VTAM provides a number of tools for verifying logons in the application program. Using these tools effectively requires planning and preparation at VTAM definition. It requires that specific procedures be established for the verification of logons in the application program, and it requires that application programs that can accept logons contain LOGON exit routines adhering to these procedures. (Though an application program can accept a logon without using a LOGON exit routine, the program would have access to the logon message only through the exit routine.)

For an application program to provide logon verification effectively, the user should:

- Establish the format and content of permissible logons for each application program. Permissible logons might be required to contain terminal-user identifications and passwords as part of the logon data.

- For some terminals, use a USS definition table to define logon formats.

- For local 3270, BSC, and start-stop terminals, use an interpret table to define logon formats to VTAM and to specify which application program is to receive notification for each logon.

- Use the VTERM, TERMINAL, LU, or LOCAL and the APPL definition statements to identify valid terminals and application programs to VTAM.

- Use the GROUP, LINE, CLUSTER, VTERM, TERMINAL, LU, or LOCAL definition statement also to specify which USS definition table or interpret table is to be used for each terminal.

- Modify, if necessary, and activate VTAM's network solicitor (for local 3270, start-stop, and BSC terminals).

Once the user has prepared the telecommunication system for terminal-initiated logons, the application program can provide a final authorization check. To assist in this final check, VTAM provides:

*A logon exit in the application program:* This exit is triggered whenever a logon is to be passed to the application program.

*A means of obtaining the logon data:* VTAM's INQUIRE macro instruction can be used to obtain the logon data. If user-defined procedures specify that the logon data contain required information (such as a user identification and a password), this information can be inspected and verified.

*A means of determining device type:* VTAM's INQUIRE macro instruction can also be used to determine the type of device from which the logon was received. If the application program is not equipped to handle such devices, the request can be rejected.

*A means of identifying the specific terminal:* Input to the LOGON exit routine includes the symbolic name of the terminal. The name is established at VTAM definition. Through user-defined procedures, valid terminal names can be made known to each application program. Criteria for accepting or rejecting logons from specific terminals can then be established by the user.

Thus, for a terminal to log on to an application program using VTAM's logon facilities, the following conditions must be met:

The terminal, the logon, and the application program must be defined at VTAM definition.

A USS definition table must be defined for each logical unit that uses character-coded logons (and logoffs).

An interpret table containing valid logon formats must be created for each local 3270, BSC, and start-stop terminal, unless the OS/VS logon is to be used.

The authorization exit routine (if supplied) must authorize the logon.

The application program must accept the logon.

VTAM also allows a user to control other types of logons. As noted above, the automatic logon is specified at VTAM definition. Although this specification can be altered by the network-operator, all connection requests can be monitored by the authorization exit routine. Also, the application program can accept or reject connection requests. Thus, even if an unauthorized connection is requested by the network operator, the request must still pass the authorization checking of the authorization exit routine and of the application program.

Application program-initiated logons are also controlled. (The application program-initiated logon results from issuing a SIMLOGON, a CLSDST with the PASS option, or an OPNDST with the ACQUIRE option macro instruction.) Authorization to issue the SIMLOGON, the CLSDST (with the PASS option), or the OPNDST (with the ACQUIRE option) macro instructions is provided at VTAM definition by the application program's APPL definition statement. In addition to this authorization, connections resulting from application program-initiated logons must pass the authorization exit routine check (if supplied) and must be accepted by the application program requested.

**Security Considerations for OS/VS Logon:** If an OS/VS logon can be issued from a local 3270, BSC, or start-stop terminal, that terminal has access to any application program in the VTAM system. If caution is not exercised when authorizing OS/VS logons, the

security of the telecommunication system may be jeopardized. A user can define data to be added to the OS/VS logon. The application program can then inspect this data to decide whether to accept or reject the logon. In addition, an OS/VS logon must also be validated by the authorization exit routine (if any). This routine could be designed to control security-sensitive connections. Therefore, by requiring a password in the logon data and by coding an authorization exit routine, a user can prohibit unauthorized use of the OS/VS logon.

### **Symbolic Names**

As a further aid in controlling connections, VTAM nodes can be addressed in the active system by symbolic names. The symbolic names are assigned at VTAM definition. Terminal names are provided by the **TERMINAL**, **VTERM**, **LU**, or **LOCAL** definition statements; NCP group names are provided by the **GROUP** definition statement; line names are provided by the **LINE** definition statement; cluster controller names are provided by the **CLUSTER** definition statement; component names are provided by the **COMP** definition statement; and application program names are provided by the **APPL** definition statement. See "The Major and Minor Node Structure" in Chapter 3 for more information on assigning symbolic names to nodes.

### ***Controlling Access to VTAM***

An application program must open an access method control block (ACB) before it can use VTAM resources and facilities. To be opened, this control block must supply an application program identification for verification by VTAM. The identification is the name of an **APPL** definition statement specified and filed in a member of the VTAM definition library during VTAM definition.

The user may also specify in the **APPL** statement that any attempt to open an **ACB** using the name of the statement as an application program identification must include a password. The password specified in the **ACB** must agree with that specified in the **APPL** definition statement if the **ACB** is to be opened.

In addition to controlling the names of authorized application programs and the specification of passwords, a user can control an application program's initial use of VTAM. A program's initial use of VTAM (that is, issuing an **OPEN** macro instruction for a VTAM **ACB**) can be restricted by controlling the activation of major nodes. Thus, even if the identification and the password match an **APPL** specification, the open request is denied if the major node containing the **APPL** specification has not been activated or if the specification is already in use by another opened **ACB** that used the same identification and password.

### ***Controlling the Use of VTAM Facilities***

Although an application program has been recognized by VTAM (a VTAM **ACB** has been successfully opened), it may still not have access to all of VTAM's facilities. The user controls access to the following facilities during VTAM definition:

- The passing of a terminal connection to another application program (**CLSDST** macro instruction with the **PASS** option)

- The initiation of a terminal connection using the **OPNDST** macro instruction with the **ACQUIRE** option or the **SIMLOGON** macro instruction

- The use of block processing instead of message or transmission processing (for terminals connected in basic mode only)

- In OS/VS2 MVS, the use of authorized path

- In OS/VS and DOS/VS Release 33 the use of the **SEND CMD** and **RCV CMD** macro instructions to enter network operator commands and receive network operator messages

Passing and acquiring enable an application program to initiate a connection request for a terminal. To help ensure that only authorized connections are made, VTAM allows the user to control connection requests initiated by application programs.



The authorization specification in the APPL definition statement provides control for block processing and application program-initiated connection. An application program can use only those facilities specifically authorized in the APPL definition statement associated with its ACB. An application program can use block processing by specifying the BLOCK option of the PROC operand of the NIB macro instruction.

*OS/VS2 System Programming Library: Supervisor* describes the specification and use of authorized path.

### **Protecting Confidential Data**

When data is transmitted between an application program and a terminal, it passes through VTAM and through NCP buffers. These buffers are obtained from and returned to common buffer pools for each transmission request. To protect confidential data, the application program can request that VTAM buffers be cleared before being returned to the buffer pools. This request is made by specifying the PROC=CONFTXT option of the node initialization block (NIB). (See "The VTAM Language" in Chapter 5 for information on the NIB.)

Buffer traces of confidential data produces only the name of the application program, the name of the terminal, and the direction of the data flow. The confidential data is not included in the trace records. (A buffer trace of nonconfidential data includes the data.) The PROC=CONFTXT option is also used by VTAM's trace facility to define confidential data.

## **Other Telecommunication Access Methods**

VTAM can coexist with QTAM and BTAM under DOS/VS and with BTAM and TCAM under OS/VS. QTAM programs, BTAM programs, and TCAM programs that do not use the communications controller in network control mode can be executed concurrently as long as they have separate telecommunication networks. Additionally, when VTAM and TCAM are both in the operating system, TCAM programs that use terminals attached to a communications controller in network control mode are supported through VTAM (except in OS/VS2 SVS). Figure 7-7 shows an OS/VS telecommunication system with TCAM and VTAM being executed concurrently.

**Note:** *Starting with Release 10 of TCAM, TCAM application programs can communicate directly (that is, without going through VTAM) with terminals attached to communications controllers operating in network control mode. For information on that kind of TCAM communication, see OS/VS TCAM Concepts and Applications, GC30-2049. The information in the following sections and in other parts of this manual applies to releases of TCAM prior to Release 10.*

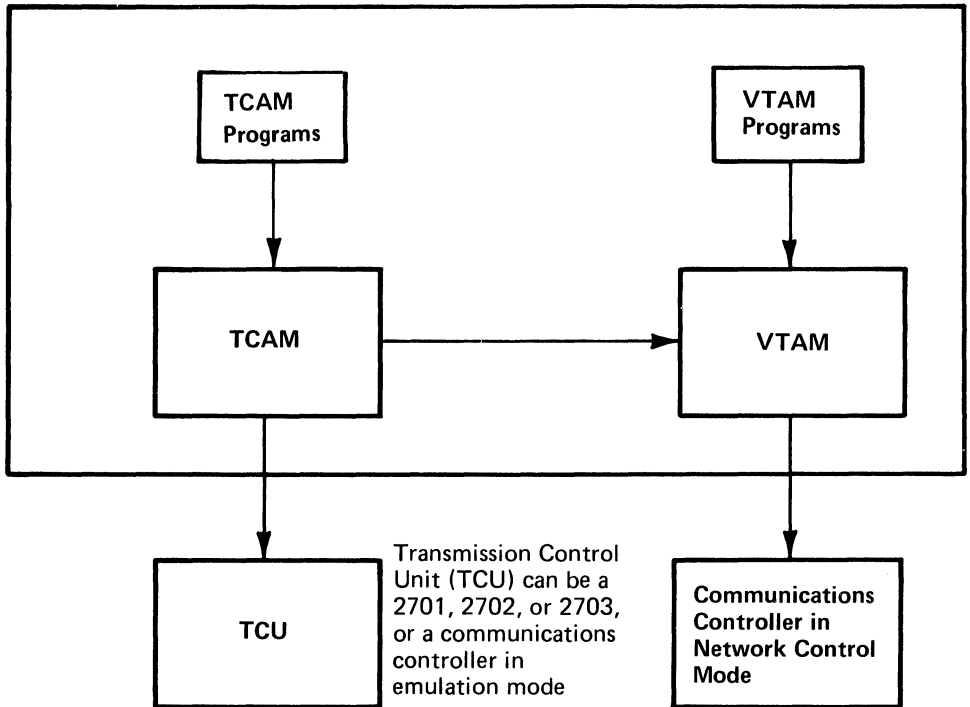
### **TCAM Programs under VTAM**

TCAM application programs and the message control program (MCP) can share the resources of a VTAM telecommunication network with application programs written for VTAM (except in OS/VS2 SVS). When sharing a network with TCAM, VTAM processes requests for all remote terminals attached to a communications controller in network control mode and, optionally, requests for local 3270s. TCAM supports terminals attached to other transmission control units, including those terminals attached to a communications controller in emulation mode (with or without PEP), and local devices. The TCAM user can choose between having individual local 3270 devices supported directly or through VTAM.

The principal advantages of using TCAM through VTAM are:

- The ability to share network resources between VTAM and TCAM programs
- The ability to allow terminals to log on to TCAM
- The availability of the queued-control capability of TCAM in a shared system

**Host Computer**



VTAM and TCAM programs use VTAM to communicate with terminals in the VTAM network. TCAM programs can also communicate with terminals attached to a 2701, 2702, or 2703 transmission control unit, or through a communications controller in emulation mode. The same communications controller can be shared by TCAM programs using emulation mode and VTAM programs using network control mode.

Figure 7-7. Communications Controllers and Transmission Control Units in a Telecommunication Network

Existing TCAM application programs may not require changes, recompilation, or reassembly; their interface with the TCAM MCP remains the same. However, application programs that use TCAM operator control commands should be evaluated to ensure that they will operate as expected in the new environment.

The following TCAM macro instructions are altered for VTAM operation: CODE, INTRO, MSGFORM, MSGGEN, STARTMH, and TERMINAL. An MCP that uses any of these macro instructions must be reassembled.

TCAM in a shared VTAM and TCAM environment depends upon VTAM to exercise physical control over stations and lines attached to a communications controller in network control mode. Differences exist between OS/VS TCAM Release 5, which directly supports the communications controller in network control mode, and the version of TCAM that operates in a shared environment with VTAM. These differences consist of:

- TCAM functions not available in the shared VTAM and TCAM environment
- TCAM operator control functions that have an altered meaning in a shared VTAM and TCAM environment
- TCAM operator control functions that are replaced with similar VTAM functions
- TCAM operator awareness messages that are replaced with VTAM messages routed to the VTAM network operator rather than to a TCAM operator control station

### TCAM Functions Not Available in a Shared Environment

Certain functions supported by OS/VS TCAM Release 5 for the network control program are not available in the shared network. Some of these are partially replaced with VTAM functions, while others are not. The functions that are not available are listed below.

- Modifications to dial digits, polling characters, and addressing characters made with ICHNG and TCHNG macro instructions are not preserved by a warm start of the NCP. These changes can be reestablished after a restart by a user-written TCAM application program that maintains a record of changes and make the changes based on operator notification that the NCP has been restarted.
- If the user is using NCP/VS Version 4, Modification Level 1 or a subsequent level, the following items are preserved by VTAM for a warm start of the NCP: line and terminal status, service-seeking pause, session limit, negative-response limit, block-handler sets, and transmission limit. TCAM's checkpoint/restart of its MCP is maintained as with TCAM Release 5. If the user is using a version of NCP/VS earlier than 4.1, none of the items listed above is preserved for a warm restart of the NCP.
- The TCAM operator command to change the dial mode of a switched line between manual and automatic dial (Change Dial Mode) for lines connected to a communications controller is not supported. VTAM allows the user to specify the dial mode of a switched line during network generation.
- The TCAM operator command to set the NCP time and date (Set 3705 Time and Date) is no longer supported. VTAM provides this function when the NCP is loaded.
- The TCAM operator command to display 32 contiguous bytes of communications controller storage (Display 3705 Storage) is no longer supported.
- The TCAM Release 5 capability to designate a communications controller as a backup and then switch dynamically to this backup in the event of controller failure, using a warm start, is not provided. The operator commands involved (Activate 3705 Backup, Switch 3705 Backup, Switch 3705s) are no longer supported. The VTAM user can manually switch between two communications controllers with appropriate physical switching equipment and the use of VTAM's VARY command.
- The TCAM Release 5 capability to dynamically switch a communications controller through a second Type 2 Channel Adapter to a backup CPU (with the Switch 3705 Channel Adapter operator command) is not supported in the VTAM/TCAM shared network. The VTAM user can manually switch to a backup CPU and reactivate the NCP with the VARY command. Operator-initiated switching through a second channel to the same CPU by a toggle switch on the communications controller is still available in the shared network; for more information on this feature see the *OS/VS TCAM Programmer's Guide*.
- The TCAM Release 5 operator commands for switching between specific and general polling for the 3270 Information Display System (Activate General Poll, Deactivate General Poll) are not supported for 3270 systems under the control of VTAM. VTAM always uses general polling for these stations.
- The CUTOFF and MSGLIMIT message handler macro instructions are not applicable for stations managed by VTAM. The CUTOFF operand of the NCP's LINE macro instruction can be used.
- The Read Full Buffer support available in TCAM Release 5 for local 3270 Information Display Systems is not available for local 3270 stations managed by VTAM. Users who require this support, described in the *OS/VS TCAM Programmer's Guide*, should use the IOS local support option available with TCAM.
- The input data from a remote 3270 managed by VTAM does not contain the control unit or station addresses. The input format is the same for the local 3270 and remote 3270. (The output format is unchanged.)

### **Altered TCAM Operator Control Functions**

Certain TCAM Release 5 operator command functions are modified in the VTAM/TCAM shared network environment. In the shared network, VTAM exercises physical control over the network, and certain TCAM functions which previously permitted dynamic physical reconfiguration are now limited to TCAM logical reconfiguration. The operator commands involved are:

- Activate Station to Receive and Transmit
- Activate Station to Transmit
- Deactivate Station for Receive and Transmit
- Deactivate Station for Receive
- Start Line Transmission
- Stop Line Transmission
- Suspend Transmission
- Release Intercepted Station

The altered TCAM application program macro instructions are HOLD and MRELEASE.

For lines and stations associated with TCAM through VTAM, these commands and macro instructions are effective only for message traffic that is being handled by TCAM. If a line or terminal in the shared network is used only by TCAM, the operator command or macro instruction has the same effect as in TCAM Release 5. If, however, the resource is shared, data that is not handled by TCAM can still reach a station for which data flow has been inhibited by a TCAM function.

VTAM provides network operator commands for activating and deactivating terminals and lines. These commands can be used to prevent all data flow to or from a station.

The TCAM Release 5 operator commands that display which lines or stations are active or inactive are still available in a shared network, but they display only the stations and lines activated or deactivated by TCAM for TCAM data. These commands do not display the status of lines and stations activated or deactivated by VTAM commands. These commands are as follows:

- Display Active Stations
- Display Station Status and Message Numbers
- Display Intercepted Stations
- Display Inactive Line Entries
- Display Inactive Open Lines
- Display Line Status and Message Error Record

### **TCAM Operator Control Functions Replaced by VTAM Operator Control Functions**

Certain TCAM Release 5 operator control functions are replaced in the VTAM/TCAM shared environment by VTAM network operator functions which are not available from TCAM operator control stations. The replaced TCAM operator functions are:

- Display 3705 Status
- Activate a 3705
- Deactivate 3705 Line and Terminal
- Dump 3705 Storage
- IPL a 3705
- PEP Switch Line Mode

Start/Stop BTU Trace  
Change NCP Load Module  
Change Session Limit  
Change 3705 Transmission Limit  
Activate/Deactivate Line Trace  
Change Polling Delay Duration

**TCAM Operator Messages  
Replaced with VTAM  
Operator Messages**

In the VTAM/TCAM shared network, VTAM awareness messages replace the TCAM operator awareness messages present in OS/VS TCAM Release 5. TCAM may route the Release 5 operator awareness messages to operator control stations, but the VTAM awareness messages that replace them are directed to the VTAM network operator. Awareness messages deal with:

Channel operations  
NCP status  
Line and terminal errors

Replies to TCAM operator commands continue to be routed to the TCAM operator control stations that enter the commands. Additionally, the TCAM ERRORMSG message handler macro instruction can be used to route information to TCAM application programs.

*Note: The following TCAM operator commands and macro instructions still exercise some degree of physical control over the shared network, and their effect on the VTAM portion of the network should be considered before they are issued:*

Change 3705 Line Speed.  
Switch 3705 Devices (dial backup for non-SDLC leased lines). If using this TCAM feature, do not use VTAM operator commands to activate or deactivate the lines to these devices.  
TCHNG  
ICHNG

**Other  
Telecommunication  
Access Methods in  
DOS/VS**

In a DOS/VS system, QTAM, BTAM, and VTAM can operate concurrently. QTAM and BTAM programs do not interact with VTAM. QTAM programs use QTAM, and BTAM programs use BTAM to communicate with:

Terminals attached to transmission control units  
Terminals attached to communications controllers by lines in emulation mode  
Terminals attached locally (BTAM only)

VTAM application programs use VTAM to communicate with:

Terminals attached to communications controllers by lines in network control mode  
3270 terminals attached locally  
3790 terminals attached locally

Lines attached to communications controllers using the NCP with PEP can be used in either network control or emulation mode with an appropriate access method.

Figure 7-8 illustrates concurrent use of QTAM, BTAM, and VTAM in DOS/VS.

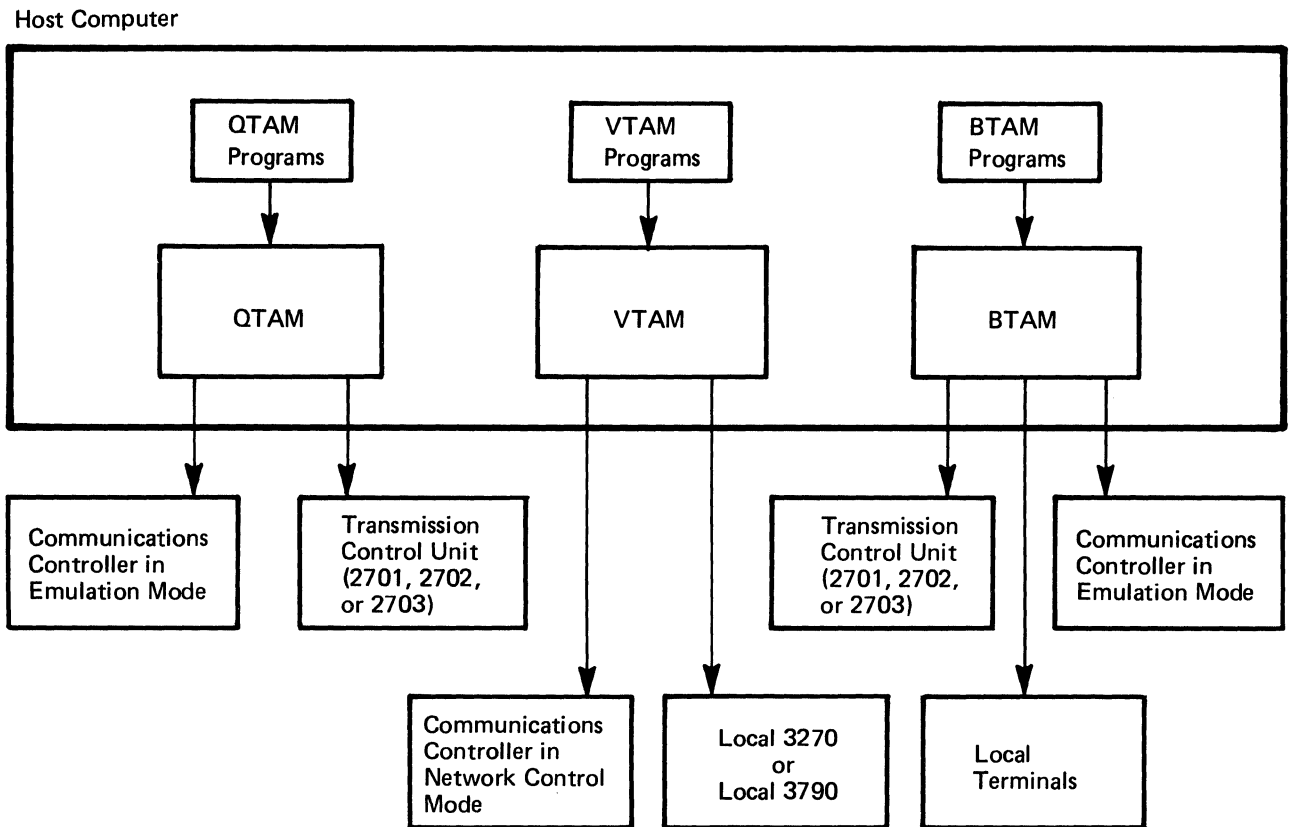


Figure 7-8. Other Telecommunication Access Methods in DOS/VS

With concurrent execution of the access method, a single application program can use both BTAM and VTAM to communicate with separate networks, provided that all requirements of both access methods are met.

**Other  
Telecommunication  
Access Methods in  
OS/VS**

In an OS/VS system, BTAM, TCAM, and VTAM can operate concurrently.

BTAM programs use BTAM, and TCAM programs use TCAM to communicate with:

- Terminals attached to transmission control units
- Terminals attached to communications controllers by lines in emulation mode
- Terminals locally attached (except 3790 terminals)

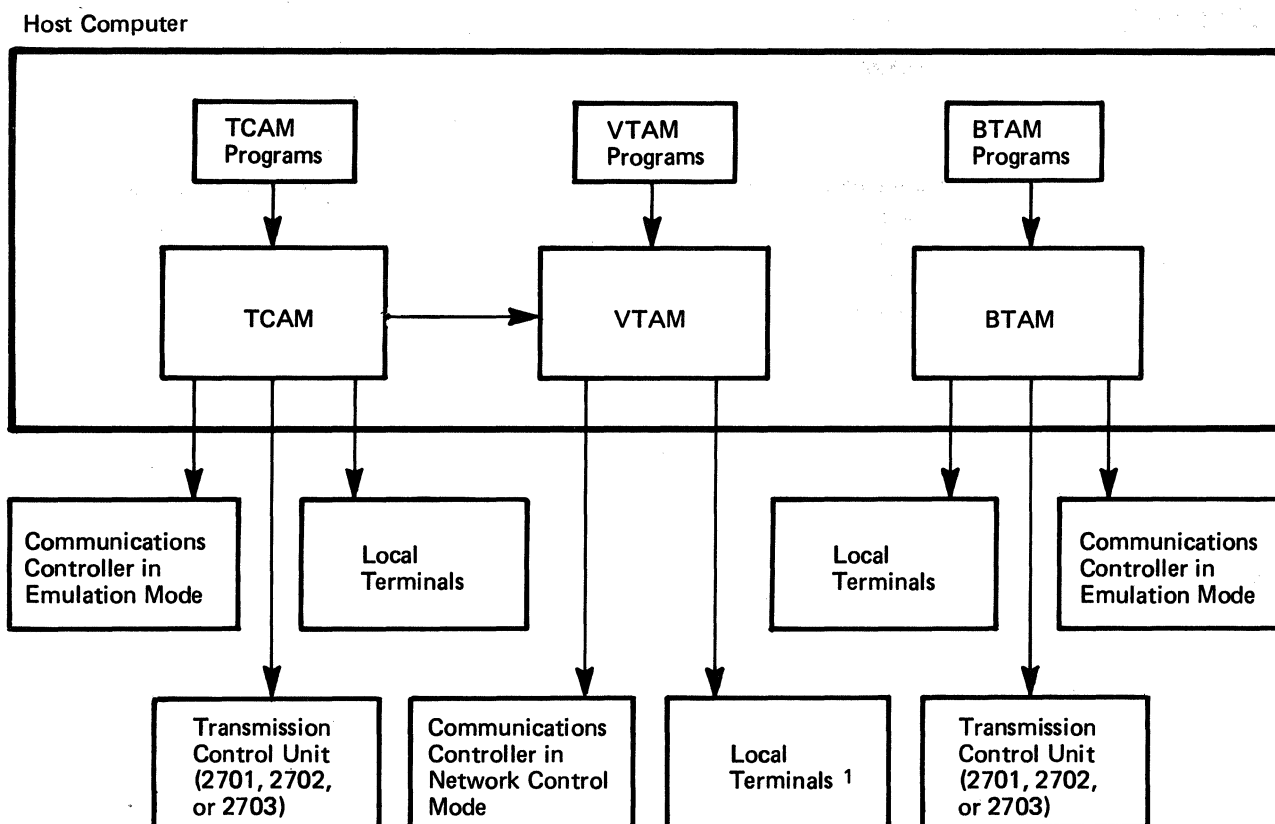
TCAM programs use VTAM to communicate with:

- Terminals attached to communications controllers by line in network control mode
- 3270 terminals locally attached

**Note:** TCAM programs can communicate with local 3270s either directly or through VTAM.

VTAM application programs use VTAM to communicate with:

- Terminals attached to communications controllers by line in network control mode
- 3270 terminals locally attached
- 3790 terminals locally attached



<sup>1</sup> VTAM programs can communicate locally with 3270 and 3790 terminal systems. TCAM programs using VTAM can communicate locally with 3270 terminal systems.

Figure 7-9. Other Telecommunication Access Methods in OS/VS

Lines attached to communications controllers containing the NCP with PEP can be used in either network control or emulation mode with the appropriate access method. Figure 7-9 illustrates the concurrent use of BTAM, TCAM, and VTAM in OS/VS.

With the concurrent execution of the access methods, a single application program can use both BTAM and VTAM to communicate with separate networks, provided that all of the requirements of both access methods are met. In addition, an application program can use both VTAM and TCAM; in this case, the networks can be separate or the same.

## CHAPTER 8. SUPPORT FOR LOCAL 3270, BSC, AND START-STOP TERMINALS

In addition to its support for terminals in a Systems Network Architecture (SNA) environment, VTAM supports the local 3270, BSC, and start-stop terminals listed in Appendix A. The information in the previous chapters applies to both support of SNA terminals and support of local 3270, BSC, and start-stop terminals, except as noted. This chapter tells what specific facilities or requirements are not applicable to local 3270, BSC, and start-stop terminals and describes the special facilities and requirements for these terminals that are not described elsewhere in this book.

Before deciding whether to use VTAM for these terminals, a user should consider using BTAM. The user may wish to continue to use existing BTAM programs, or may wish to modify these programs to interface with new VTAM application programs, or may wish to use BTAM macro instructions in new programs.

### Using BTAM

Figure 8-1 shows the major similarities and differences between VTAM application programs and BTAM application programs. VTAM application program characteristics shown in Figure 8-1 are discussed in more detail in this chapter and in Chapter 3.

Figure 8-2 shows that communication using BTAM can be combined with communication using VTAM in a number of ways:

An application program that uses VTAM record mode macro instructions to communicate with logical units can also include BTAM macro instructions to communicate with local 3270, BSC, and start-stop terminals.

An application program that uses VTAM record mode macro instructions to communicate with logical units can use VTAM basic mode macro instructions to communicate with some local 3270, BSC, and start-stop terminals and can also use BTAM macro instructions to communicate with other local 3270, BSC, and start-stop terminals. A program of this nature would be required to communicate with some local 3270, BSC, and start-stop terminals that were supported by VTAM and some that were not supported by VTAM.

One application program can be used for VTAM communications and another can be used for BTAM communications.

A user that wants to continue to use terminals not supported by VTAM would have to use one of these combinations. However, even if all local 3270, BSC, and start-stop terminals in a network are supported by VTAM, the user might want to use BTAM to communicate with them.

### Using VTAM

As for SNA terminals, using VTAM for local 3270, BSC, and start-stop terminals requires:

Creating a VTAM network (that includes these terminals)

Operating the network

Writing VTAM application programs

Except as described below, the information in Chapters 3, 4, and 5, on creating a network, operating the network, and writing VTAM application programs apply for local

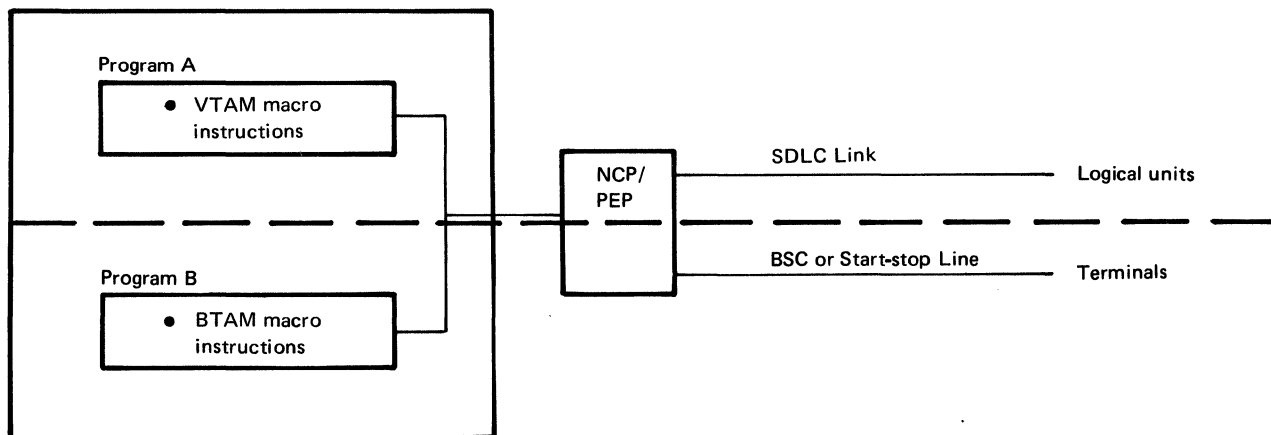
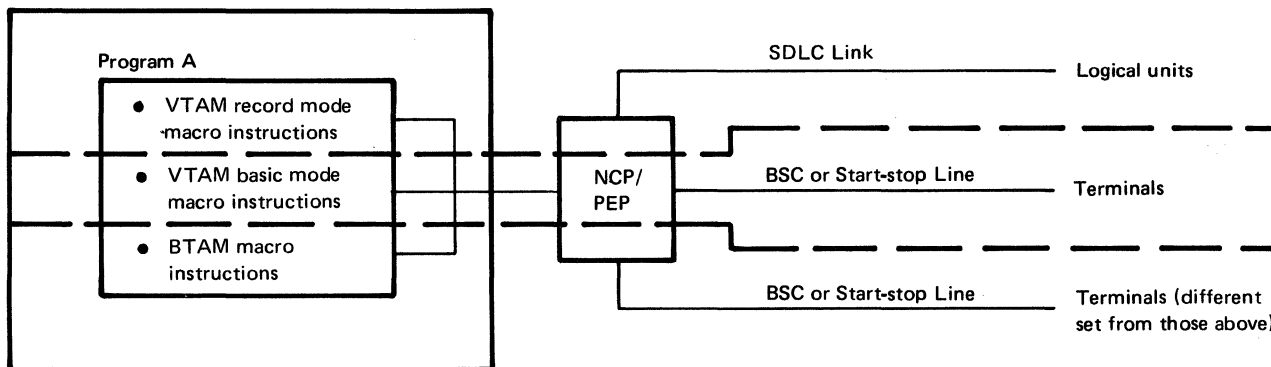
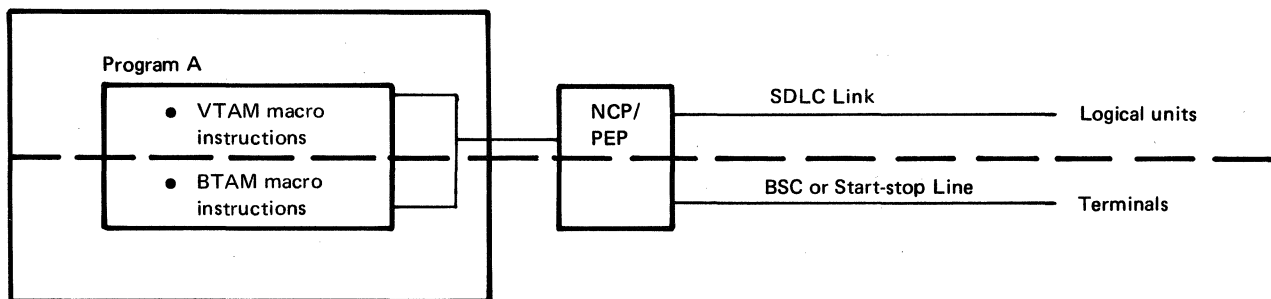


Characteristic	VTAM Application Program	BTAM Application Program
<p><i>General Characteristics of the Access Method</i></p> <p>Terminal-sharing</p> <p>Line-sharing</p> <p>Line-scheduling logic required</p> <p>Polling required on part of application program</p> <p>Exit routines scheduled for special conditions such as a logon request</p> <p>Local/remote 3270 handling</p> <p>Primary type of terminal communication</p> <p>Direct-control or queued access method</p>	<p>Yes</p> <p>Yes</p> <p>No</p> <p>No</p> <p>Yes</p> <p>Can be communicated with in same mode as logical units; no coding distinction between local and remote</p> <p>Communicates primarily with logical units (program logic). Can also communicate with local 3270 start-stop, and BSC terminals</p> <p>Direct-control</p>	<p>No</p> <p>No</p> <p>Yes</p> <p>Yes</p> <p>No</p> <p>Requires different coding for local 3270 and remote 3270</p> <p>Communicates primarily with start-stop and BSC terminals</p> <p>Direct-control</p>
<p><i>Language Characteristics</i></p>	<p>VTAM macro instructions</p> <p>Keyword operands</p> <p>Macros common for DOS/VS and OS/VS</p> <p>Two sets of I/O macro instructions: record-mode (RECEIVE-SEND) for logical units and 3270 terminals, and basic-mode (READ-WRITE) for local 3270, BSC, and start-stop devices</p> <p>Destination-oriented</p>	<p>BTAM macro instructions</p> <p>Positional operands</p> <p>Macros different for DOS/VS and OS/VS</p> <p>One set of I/O macro instructions</p> <p>Line-oriented</p>
<p><i>Program Organization</i></p>	<p>Telecommunication normally separate from processing</p> <p>Can have VTAM post an ECB or schedule an exit-routine when I/O event completes</p>	<p>Telecommunication normally separate from processing</p> <p>BTAM posts an ECB when I/O event completes</p>

Figure 8-1 (Part 1 of 2). Major Similarities and Differences Between VTAM and BTAM Application Programs

Characteristics	VTAM Application Program	BTAM Application Program
<i>Functions Provided</i>		
Define line groups	All resources are defined during VTAM definition	Use DCB or DTFBT macro instruction
Define terminals	All resources are defined during VTAM definition	Use DFTRMLST macro instruction
Initialize program	Use OPEN macro instruction	Use OPEN macro instruction
Connect terminals dynamically to application program	Use OPNDST macro instruction	Not available, because terminals are statically connected when application's job step is initiated
Release control of a terminal to another program that requests connection to it	When requested, use CLSDST with OPTCD=RELEASE specified	No comparable function
Receive input	Use RECEIVE or READ macro instruction	Use READ macro instruction
from any connected terminal	Specify input can be from any terminal	Must poll each line separately
from a specific terminal	Specify input is to be received from a specific terminal	Specify terminal list entry
Send output	Use SEND or WRITE macro instruction	Use WRITE macro instruction
Have data scheduled for output from access method buffers (output buffering)	Specify that the data is to be scheduled for output	No comparable function
Test, display, or modify control block fields	Use manipulative macro instructions: TESTCB, SHOWCB, MODCB. Or use assembler language instructions	Use assembler language instructions
Record transmission errors	Performed by NCP and VTAM	Use error recording macro instructions

Figure 8-1 (Part 2 of 2). Major Similarities and Differences Between VTAM and BTAM Application Programs



**Notes:** Local and remote 3270 terminals can be communicated with using VTAM record mode macro instructions, VTAM basic mode macro instructions, or BTAM macro instructions. Using VTAM record mode allows logical units and 3270s to be communicated with using the same set of macro instructions.

Figure 8-2. Using BTAM and VTAM to Communicate with Local 3270, BSC, and Start-Stop Terminals

3270, BSC, and start-stop terminals. Chapters 6 and 7 also apply, except where noted, to local 3270, BSC, and start-stop terminals. In addition, VTAM provides special support consisting of:

- A network solicitor that controls logons

- Interpret tables that can be defined to translate a name in a logon to the name of a VTAM application program

- Support for start-stop and BSC terminals on switched lines

- A set of basic mode macro instructions—SOLICIT, READ, WRITE, RESET, and others—used when writing a VTAM application program

Because the BSC and local 3270s can also use the record mode communication macro instructions that are used for SNA terminals (with certain restrictions), this support is also discussed in this chapter.

## Topics Not Applicable to Local 3270, BSC, and Start-Stop Terminals

In Chapter 3, “Creating a VTAM Telecommunication System,” these topics do not apply to local 3270, BSC, and start-stop terminals:

- “Defining Terminal-Initiated Connection” (which includes defining USS definition tables and logon mode tables). Instead, see “Defining Terminal-Initiated Logons for Local 3270, BSC, and Start-Stop Terminals” in this chapter.

- “Defining Local SNA Major Nodes” and “Defining Switched SNA Major Nodes.” These topics apply only to SNA terminals.

In Chapter 4, the topics “Activating and Deactivating Local SNA Major Nodes” and “Activating and Deactivating Switched SNA Major Nodes” do not apply.

In Chapter 5, the topic dealing with establishing session parameters under the topic “Connection” does not apply. The communication macro instructions—SEND, RECEIVE, RESETSR, and SESSIONC—and the topic “Record-Mode Communication” do not apply to communication with local 3270, BSC and start-stop terminals (unless using record mode support to communicate with local and BSC 3270 terminals). Instead, see “Communicating with Local 3270, BSC, Start-Stop Terminals” in this chapter. See *VTAM Macro Language Guide*, for examples of program logic using the basic mode macro instructions. See *VTAM Macro Language Reference*, for specific terminal considerations when writing a VTAM application program to communicate with one or more types of local 3270, BSC, and start-stop terminals.

## Creating a VTAM System That Includes Local 3270, BSC, or Start-Stop Terminals

This process consists of generating a network control program (unless only local terminals are in the network) and defining the network configuration and characteristics to VTAM, as described in Chapter 3. In defining connection procedures, the user may want to understand and define the use of a network solicitor. The user can also use interpret tables to have VTAM interpret a logon from a local 3270, BSC, or start-stop terminal.

GROUP, LINE, CLUSTER, TERMINAL, COMP, and VTERM statements that define the network to VTAM can specify:

- Automatic logon and interpret table requirements.

- A description of the features for a BSC 3270 terminal.

The initial status of a terminal or a cluster control unit when the NCP is activated by VTAM.

The buffer limit for a terminal. (See "Defining VTAM Buffering" in Chapter 3 for a description of how buffer limits are established.)

The name of each terminal. The name of the terminal is usually the name of the **TERMINAL** or **COMP** statement. If the **TERMINAL** statement defines a *logical connection terminal* (that is, if it contains the **CTERM=YES** operand), the name of the terminal must be specified in an additional operand, **UTERM**. The name specified by **UTERM** is used only by VTAM and applies to terminals on a switched line. See "Defining a Switched Network for Start-Stop and BSC Terminals" in this chapter for an explanation of the use of the **UTERM** name. Refer to the *NCP Generation* publication for an explanation of the **CTERM** operand.

The name of each group, line, and cluster control unit (if any). Names are specified in the **GROUP**, **LINE**, and **CLUSTER** statements.

### ***The Network Solicitor***

The network solicitor monitors local 3270, BSC, and start-stop terminals for logons and passes the terminals with valid logons to the appropriate application program. Using the network solicitor, a user can permit terminal-initiated logons for local 3270, BSC, and start-stop terminals.

A version of the network solicitor is automatically included in VTAM during system generation. This network solicitor has the name **NETSOL** and can release terminals to requesting application programs. It can also be started and stopped by VTAM's network operator facilities.

The user can retain this network solicitor, modify it, or replace it. If the VTAM network solicitor is to be used, the user need only establish logon capabilities for local 3270, BSC, and start-stop terminals as described in "Defining Terminal-Initiated Logons for Local 3270, BSC, and Start-Stop Terminals" later in this chapter. The network solicitor can be modified through the use of VTAM's **NETSOL** macro instruction as explained below.

### **How the Network Solicitor Works**

Figure 8-3 shows how the network solicitor functions. The network solicitor monitors terminals assigned to it by the automatic logon specification. Terminals are monitored only when they are active but not connected, or queued for connection, to an application program.

When a terminal being monitored by the network solicitor enters a message, the network solicitor determines whether the message is a valid logon. The logon is validated in one of two ways:

If an interpret table is specified for the terminal, a search is made in that table for an entry corresponding to the logon.

If no interpret table is specified and the operating system is OS/VS, the logon is checked for an OS/VS-defined format.

See "Defining Terminal-Initiated Logons for Local 3270, BSC, and Start-Stop Terminals" later in this chapter for a description of defining valid logons.

If the logon is valid, the terminal is passed to the appropriate application program, if the application is active and accepting logons. The application program is the one specified for that logon in the interpret table, or in the case of an OS/VS logon, it is the application program named in the logon itself.

Terminal operator enters logon from active local 3270, BSC, or start-stop terminal.

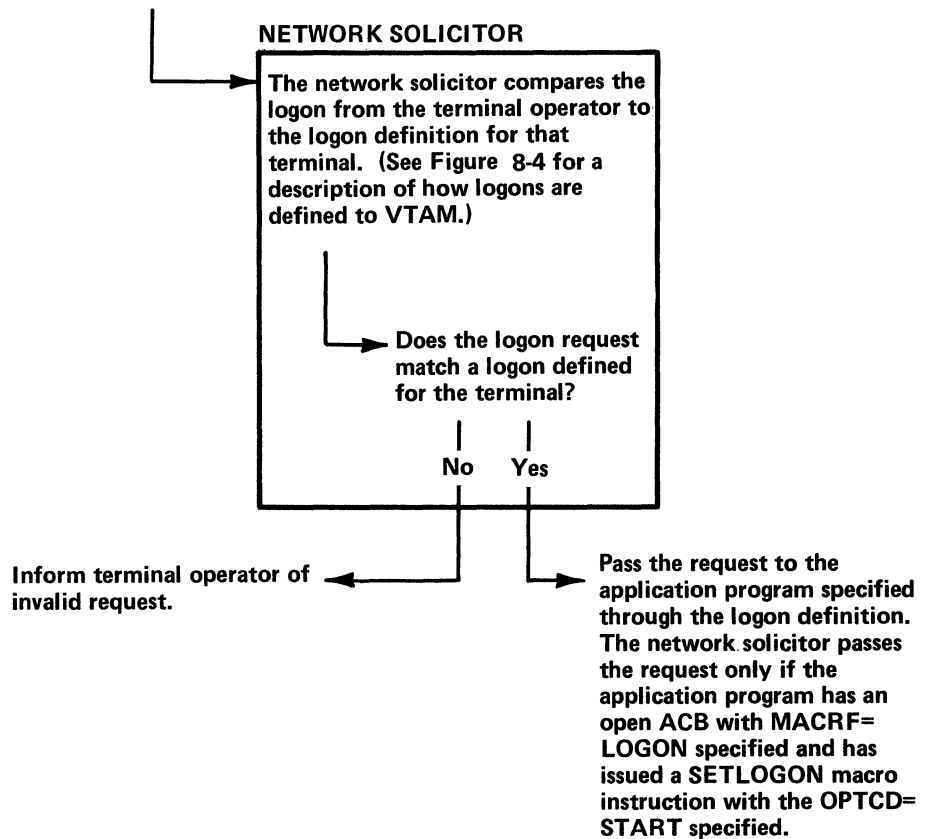


Figure 8-3. Processing a Terminal-Initiated Logon with the Network Solicitor

If the logon is invalid and the application program is not active, or if the application program is not accepting logons, the terminal operator is notified that the logon has been rejected and is invited to enter another logon.

### Modifying the Network Solicitor

The network solicitor is modified by coding, assembling, and link-editing VTAM's NETSOL macro instruction. If a NETSOL macro instruction is not used to replace the network solicitor, VTAM's network solicitor remains available for use. The modified network solicitor can replace, or be used in addition to, the IBM-supplied network solicitor. In DOS/VS, the modified network solicitor should be cataloged in the same library as the one it replaces. In OS/VS, if the modified network solicitor is to be a replacement, it must be link-edited with the VTAM modules in the VTAM load module library.

**Network Solicitor's Name:** The name of the IBM-supplied network solicitor is NETSOL. If the modified network solicitor is to replace the IBM-supplied network solicitor, this name can be retained. (Note: Only the network solicitor that runs in VTAM's partition or private address space can use the name NETSOL.) If any other name is specified, the modified network solicitor is treated as an application program by VTAM. That is, the network solicitor must run in its own partition or private address space, the user must supply an APPL definition statement for it, and it must be started and stopped like an application program. VTAM's start options and MODIFY command cannot be used to start or stop a network solicitor with a name other than NETSOL. See Chapter 4 for using the start options or the MODIFY commands with the network solicitor. The name

specified on the NETSOL macro instruction is the name in the automatic-logon specification for each terminal to be monitored by the network solicitor. The load module name of the network solicitor is ISTNSCOO; this load module name must be used when modifying the IBM-supplied network solicitor.

**Network Solicitor Messages:** The network solicitor issues a message if any of the following conditions is encountered:

The application program specified in the logon is unavailable for logons. The application program is unavailable if it is inactive, closing down, or not accepting logons.

The logon is invalid; that is, it does not match any entry in the specified interpret table or (for OS/VS only) it is not in the OS/VS-defined format.

No interpret table is specified (DOS/VS only) or no interpret table is available for the terminal. (Interpret tables are discussed later in this section.)

The telecommunication system is closing down.

An input error is encountered.

The logon is rejected by the authorization facilities of VTAM.

The terminal is not supported by the network solicitor.

For each of these conditions, the user can replace the text of the IBM-supplied messages.

**Network Solicitor Release Request:** If a user authorizes application programs to *acquire* terminals, the network solicitor should be able to release, upon request, terminals it is monitoring. If release request is specified in the NETSOL macro instruction, the network solicitor is generated with a RELREQ (release request) exit routine like the one in the IBM-supplied network solicitor. Whenever this exit routine is scheduled, the network solicitor releases the requested terminal unless a logon is being processed. (See "Acquisition" in Chapter 5 for details on the RELREQ exit routine, how it is invoked, and on acquiring terminals.)

**Password:** If the user wants the ACB for the modified network solicitor to contain a password, this password must be specified in the NETSOL macro instruction. If a password is specified, the modified network solicitor is treated as an application program by VTAM; that is, it must run in its own partition or private address space, the user must supply an APPL definition statement for it, and it must be started and stopped like an application program. VTAM's start options and MODIFY command cannot be used to start or stop a network solicitor with a password.

## Replacing the Network Solicitor

If a user does not want to use the IBM-supplied network solicitor, but does want a general-purpose terminal-initiated logon facility for local 3270, BSC, or start-stop terminals, the user can code an application program to perform the network-solicitor functions. Such an application program monitors terminals for logons and passes valid logons to the appropriate application programs. The monitoring program is treated like an application program by VTAM; an APPL definition statement has to be filed for it, and it has to be activated and deactivated as an application program. As an application program, the monitoring program still has access to the interpret tables through the INTRPRET macro instruction.

## Specifying Interpret Tables

VTAM's INTAB, ENDINTAB, and LOGCHAR macro instructions are used to construct interpret tables (Figure 8-4). The LOGCHAR macro instruction describes the text and format of a single logon. The INTAB and ENDINTAB macro instructions define an *interpret table*, which contains one or more such logons. Each interpret table must be assembled and filed separately (as a member in OS/VS or a book in DOS/VS) in the VTAM load module library. The name assigned to the member or book is the name assigned (by the INTAB macro instruction) to the interpret table. Thus, the INTAB and the ENDINTAB macro instruction define a group of logon definitions and provide a name for that group. The LOGCHAR macro instruction describes a specific logon and can be used to indicate the following:

- Whether the logon is requested by a character string or by a 3270 program function key.

- Which program function key, if any, can make the request.

- What character string, if any, can make the request. The characters specified in the LOGCHAR macro instruction are the only characters to be checked by VTAM at the beginning of the logon. The actual logon entered from the terminal can contain additional logon data to be used, for example, for password protection or accounting by the application program. This additional logon data must not be specified in the LOGCHAR macro instruction.

- The name of the application program to receive this logon.

For each logon, the user can specify in the LOGCHAR macro instruction either the name of an application program or the name of a routine (a logon-interpret routine) that is to determine the appropriate application program. All logon-interpret routines specified in the same interpret table must be link-edited with that interpret table.

The INTRPRET macro instruction allows access to the contents of the interpret table. The network solicitor uses the INTRPRET macro instruction to validate logons. (The macro instruction can be used similarly by application programs.) The following is a description of the network solicitor's use of the INTRPRET macro instruction and of the interpret tables.

The network solicitor invokes INTRPRET, specifying a logon received from a terminal and the name of that terminal. INTRPRET then determines if there is an interpret table for that terminal. If there is no table for that terminal, INTRPRET indicates this to the network solicitor. If there is a table, INTRPRET checks for a match between the logon passed to it and one defined by a LOGCHAR macro instruction for the table. If no match is found, INTRPRET informs the network solicitor that the logon is not in the table.

If a match is found, INTRPRET determines whether an application program or a logon-interpret routine is specified in the LOGCHAR macro instruction. If an application

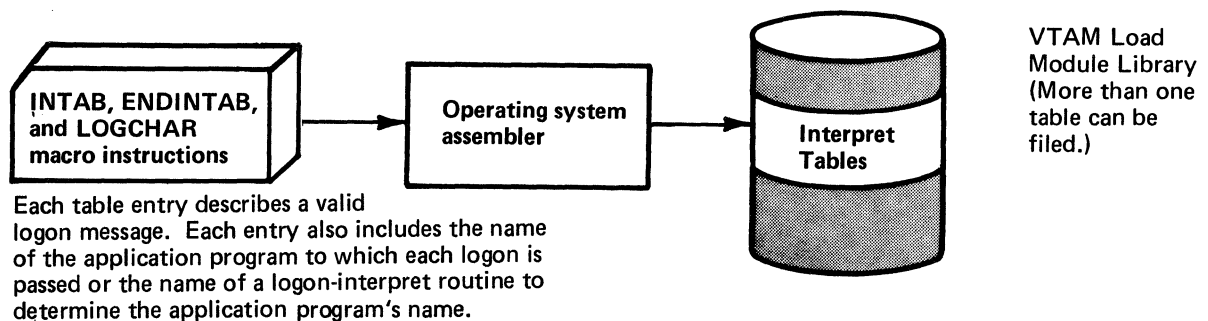


Figure 8-4. Filing Interpret Tables



program is specified, INTRPRET returns the name of the program to the network solicitor. If a logon-interpret routine is specified, INTRPRET invokes the routine, passing the logon text and terminal name. This routine, which is user written, should validate the logon. The logon-interpret routine should specify the name of the application program to receive the logon, or it should specify that the logon is invalid. This information is returned to the network solicitor.

The entire logon sequence from the terminal is given as input to a logon-interpret routine, and it can therefore contain more data than is specified in the associated LOGCHAR macro instruction. The routine can use this additional logon data to determine the application program name. This data might also contain information such as a password that is verified by the routine.

Although the interpret tables are intended primarily for validating terminal-initiated logon, they are also available to application programs through VTAM's INTRPRET macro instruction.

### ***Defining Terminal-Initiated Logons for Local 3270, BSC, and Start-Stop Terminals***

To enable a terminal operator to issue a logon from a local 3270, BSC, or start-stop terminal, the steps are as follows:

1. Modify the network solicitor. (This step is optional.)
2. Define terminal operator logon procedures and sequences to VTAM.
3. Activate the network solicitor.
4. Activate the terminal.
5. Enter a logon to be processed by the network solicitor.

Steps 1 and 2 are done as part of VTAM definition. Steps 3 and 4 are completed by the network operator, although the degree of network-operator involvement depends upon the VTAM definition options selected. (See Chapter 4 for details on activating the network solicitor and terminals.) Step 5 is accomplished by the terminal operator.

To use the IBM-supplied network solicitor, the user must define:

1. Which terminals are to be handled by VTAM's network solicitor. (Output-only terminals cannot be handled and should not be defined.)
2. What is the format and content of each logon and what is the name of each application program to be notified for each logon.
3. Which logons can be used by each terminal.

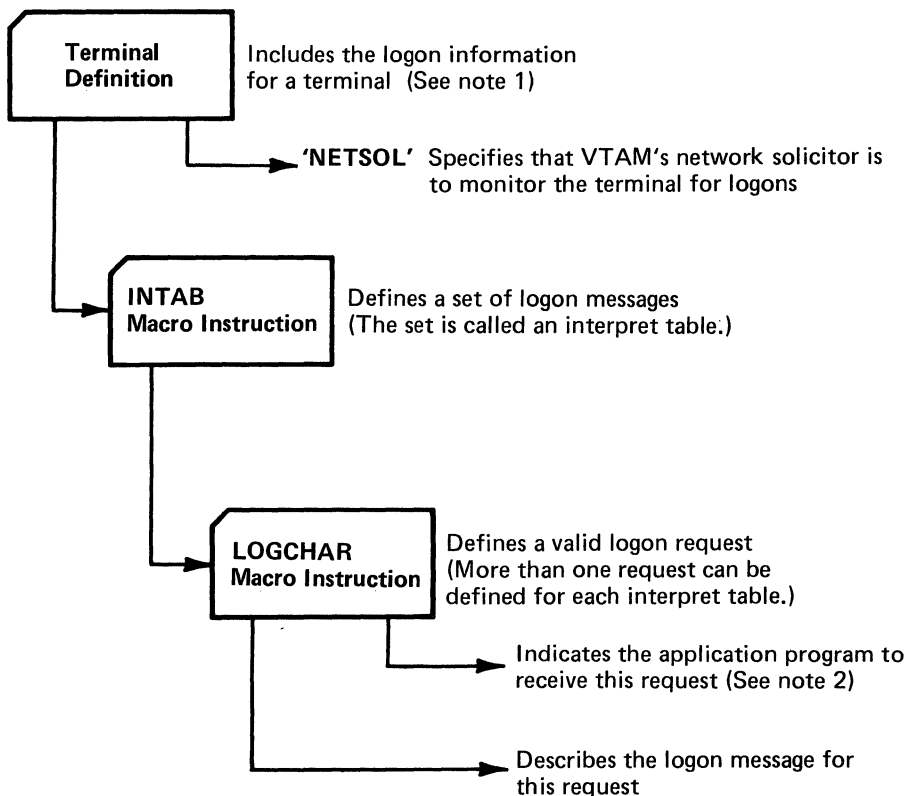
An installation can use VTAM's automatic logon capability to accomplish item 1. Instead of specifying an application program name for automatic logon in a terminal's GROUP, LINE, CLUSTER, VTERM, TERMINAL, or LOCAL definition statement, the user specifies VTAM's network solicitor. Whenever a terminal so designated is available, the network solicitor monitors it for a logon. The network operator can also temporarily assign a terminal to the network solicitor by using the VARY command.

Interpret tables define valid logon messages to VTAM and indicate which application programs are to be notified of the connection request for each valid logon (item 2). OS/VS also provides a logon that does not use an interpret table. See "Specifying Interpret Tables," earlier in this chapter, for details on setting up interpret tables.

For item 3, the interpret table used to validate logons from this terminal is named in a terminal's **GROUP**, **LINE**, **CLUSTER**, **VTERM**, **TERMINAL**, or **LOCAL** definition statement. Figure 8-5 shows how control information for processing terminal-initiated logons is defined to VTAM for local 3270, start-stop, and BSC terminals.

***Defining a Switched Network for Start-Stop and BSC Terminals***

This section describes VTAM's support for call-in, call-out, and call-in/call-out terminals. (Switched-network support is provided only for start-stop and BSC terminals as specified in Appendix A.) Also included are discussions on network operator considerations that apply to controlling switched networks. This section is meant to augment the discussion on NCP support for start-stop and BSC switched networks provided in the *NCP Generation* publication.



**Notes:**

1. The logon information can be specified in the **GROUP**, **LINE**, **CLUSTER**, **VTERM**, **TERMINAL**, or **LOCAL** statement.
2. This parameter can point to an actual application program or to a logon-interpret routine that determines the application program to receive the request.

Figure 8-5. Providing Control Information for Processing Logon Requests from Local 3270, BSC, and Start-Stop Terminals

## Call-In Terminals

Call-in (dial-in) terminals have VTAM-definition requirements that affect some network operator and application program activities. Understanding these effects requires an understanding of VTAM's support of call-in terminals.

VTAM uses the concept of a port to support call-in terminals. As noted in the *NCP Generation* publication, each call-in line must have a `TERMINAL` statement with a `CTERM=YES` operand. These statements represent logical connections to the NCP, but they represent ports to VTAM.

For VTAM to accept a call-in request over a switched line, the port for that line (in addition to the other nodes in the path) must be active. Ports are activated automatically when an NCP is loaded and activated by VTAM. Thereafter, ports can be activated or deactivated using the network operator's `VARY` command.

The name of a port is the name of the `TERMINAL` statement with the `CTERM=YES` specification. This name is used by the network operator to address the port; it is not used by an application program, because application programs connect only to terminals. Note that for a switched line, a port is a minor node; it is defined to VTAM and can be addressed.

The `TERMINAL` statement that defines a port may also contain a definition of a terminal. Such a statement is used as a definition of a terminal only if the terminal calling in cannot be identified and associated with another `TERMINAL` or `VTERM` statement. (This identification would be provided by either the NCP's MTA facility or by VTAM's and the NCP's BSC and TWX identification facilities.) That is, if a terminal calling in over a switched line cannot be identified, VTAM attempts to apply the terminal description in the port (`TERMINAL`) statement for that line to the terminal. Thus, information such as terminal type, automatic-logon specifications, and initial status is applied to the terminal calling in.

For the description on the port statement to be applied to the terminal, a `UTERM` operand must be specified in that statement. The name in the `UTERM` specification is the name of the terminal (any unidentified terminal) calling in over the line serviced by the port. See "Creating a VTAM System That Includes Local 3270, BSC, or Start-Stop Terminals" earlier in this chapter for a description of the `UTERM` specification on the `TERMINAL` statement.

An application program wishing to be connected to any unidentified terminal calling in over a specific line uses, as the name of the terminal, the `UTERM` name, not the name of the `TERMINAL` statement itself. Likewise, an automatic-logon specification in a port statement is applied to any unidentified terminal calling in over that line; the terminal is logged on to the program specified (either the network solicitor or an application program).

In summary, the following should be considered when planning to use VTAM's support for call-in terminals:

For MTA lines, the `VTERM` statement can be used to provide identification and logon information for each type of terminal. If a `VTERM` statement is not used, an MTA terminal is defined by the port statement (if that statement has a `UTERM` specification).

For BSC (and TWX) terminals, the `IDLIST` and the `VIDLIST` statements can be used to distribute identification responsibility between the NCP and VTAM.

For port definition, the `TERMINAL` statement with the `CTERM=YES` parameter defines a port.

For unidentified terminals, a UTERM name must be specified on a port statement if VTAM is to connect unidentified terminals calling in over the line serviced by the port.

#### **Call-Out Terminals**

VTAM has no special requirements for supporting call-out terminals. Using VTAM-definition and NCP facilities, a user can specify that a terminal is to be dialed automatically or manually by the network operator. The dial digits must be specified at NCP generation. For automatic dialing, the numbers are dialed by the NCP. For manual dialing, VTAM transmits a message (containing the dialing instructions) to the network operator.

#### **Call-In/Call-Out Terminals**

Special planning is required for terminals that both call in and call out. Such terminals might be represented twice to the NCP and to VTAM.

If a terminal can be identified during a call-in operation through BSC (or TWX) identification facilities, it is defined for both call-in and call-out operations by the same TERMINAL statement.

If a call-in terminal is unidentified or is an MTA terminal, it is defined by either the UTERM name in a TERMINAL statement or by a VTERM statement (MTA terminals only). If the same terminal can be called, it must have another TERMINAL statement defining its call-out characteristics. Thus, the terminal has two definitions and two names: one for calling in, the other for calling out. Although the call-out definition applies to a specific terminal, the call-in definition can apply to any valid, but unidentified (including MTA) terminal calling in.

A request from the network operator to activate or deactivate one of the terminal's definitions does not affect the other definition. For example, a deactivation request specifying the call-out name does not affect the terminal's ability to call in. If such a request is issued, the terminal can still be used to call in even though a call-out operation would be prohibited by VTAM.

Similar considerations apply for application programs connecting with terminals capable of both being called and calling in. If an application program were to connect with a terminal (MTA or unidentified) that has called in, it connects using the call-in name. Then, if the application program disconnects the terminal and subsequently attempts to reconnect it by dialing out, the name of the terminal specified in the call-out definition has to be used in the connection request.

#### ***Operating a VTAM System with Local 3270, BSC, and Start-Stop Terminals***

The operation of a VTAM network described in Chapter 4 includes the facilities that apply to local 3270, BSC, and start-stop terminals. In addition, the network operator can use the MODIFY command to start the network solicitor. This network solicitor must be (1) the IBM-supplied network solicitor or (2) the network solicitor that was modified with the NETSOL macro instruction (with the name NETSOL).

Activating the network solicitor causes all appropriate available terminals to be automatically logged onto it. Appropriate available terminals are terminals that are active but not connected, nor queued for connection, to another program and whose automatic-logon specification in their node definition indicates the network solicitor.

As long as the network solicitor remains active, it continues to monitor available terminals and passes valid logons to the appropriate application programs.

Deactivating the network solicitor with the MODIFY command causes the network solicitor to complete handling all terminal-initiated logons in process and to disconnect all

other terminals connected to it. No additional automatic logons are accepted by the network solicitor until it is reactivated.

## Writing a VTAM Application Program

The concepts and facilities described in Chapter 5, "Writing a VTAM Application Program," except where noted, apply for programs that communicate with local 3270, BSC, and start-stop terminals. A program that communicates with both logical units and local 3270, BSC, or start-stop terminals might be organized so that as much coding as possible is shared. Separate communication routines might be written for individual types of local 3270, BSC, or start-stop terminals. In a program with logon acceptance, VTAM, as the result of an OPNDST macro instruction, identifies the type of terminal (logical unit or local 3270, BSC, or start-stop terminal) by setting a value in the DEVCHAR field of the NIB furnished by the application program. By testing the value in this field, the program can determine the appropriate macro instructions and related logic with which to communicate with the terminal.

### *Basic-Mode Concepts*

The macro instructions and facilities used by the application program to communicate with VTAM-supported start-stop and BSC terminals and local 3270 terminals are different from those used to communicate with logical units. The two sets of macro instructions and facilities are called the *basic mode* and the *record mode*.

The basic mode set of macro instructions must be used for the start-stop and BSC terminals, and can also be used for the local and BSC 3270. The record mode set is used for logical units and can be used for the local and BSC 3270.

### **The Basic-Mode Macro Instructions**

Here is a brief description of the basic-mode communication macro instructions:

**SOLICIT:** Requests that VTAM solicit input from a specific local 3270, BSC, or start-stop terminal or from a group of terminals. Input is read into a VTAM buffer, not into the application program; a READ macro instruction is used to read the input into the application program's data area. The solicitation action caused by a SOLICIT macro instruction issued to a group of terminals continues until input has been received from every terminal in the group. Once input has been received from a terminal, the terminal must be resolicited unless, at connection, continuous solicitation of the terminal was specified.

**READ:** Requests that VTAM transfer data from a specific terminal or from any one of a group of terminals into an area in the application program. A request to read from any one of a group of terminals requires that a SOLICIT macro instruction be issued prior to the READ; a request to read from a specific terminal solicits input if a SOLICIT was not previously issued for that terminal. If data is already in a VTAM buffer as the result of a previous solicit or read operation, data is transferred immediately.

**WRITE:** Requests that VTAM transfer data from an application program to a specific terminal. The application program can also request that VTAM send certain control information to a terminal or write conversationally (write and then read from a terminal).

**DO:** Requests that VTAM perform the special I/O action associated with a specific LDO control block in the application program.

**RESET:** Requests that VTAM cancel all outstanding I/O requests to a terminal and, if necessary, reset any error locks that may have been set for the terminal to prevent output.

**CHANGE:** Requests that VTAM change information contained in a NIB. Using CHANGE quiescs I/O activity, disconnects the terminal, then reconnects the terminal. (The CHANGE macro instruction cannot be used with 3270 terminals.)

#### **The LDO Control Block**

In addition to the control blocks described in Chapter 5 in “Control Block Macro Instructions,” VTAM application programs that communicate with certain start-stop and BSC terminals can define a logical device order (LDO) control block. This control block defines a particular kind of I/O operations that is not ordinarily performed, such as writing a positive response with leading graphics to a System/3 or System/370 CPU. The operation is requested by issuing a DO macro instruction that specifies the LDO.

#### ***Communicating with Local 3270, BSC, and Start-Stop Terminals***

The unit of data exchanged in record mode operations is different from that exchanged in basic mode operations. In record mode, the units exchanged are *messages* (which include data) and *responses*. In basic mode, the unit of data is the *block*.

Blocks are delimited differently for different types of terminals. For start-stop terminals, a block ends with an EOB character; for BSC terminals, a block ends with an ETB or ETX character.

Although the application program can solicit more than a block from a terminal, a READ macro instruction can move only a block into the application program’s input area (or less, if the input area is smaller than a block). An output operation (a WRITE macro instruction) always sends one block to the terminal.

#### **Solicitation**

When the application program solicits data from a terminal, VTAM initiates whatever actions (such as polling or line preparation) are required to obtain data from the terminal and put it into VTAM buffers.

Read requests issued in specific-mode cause solicitation if no previously solicited data is in VTAM’s buffers and then move data into the application program’s input area. In contrast, read requests issued in any-mode can only move solicited data from VTAM’s buffers into the application program’s input area. The user of read requests in any-mode must therefore explicitly request solicitation. Figure 8-6 illustrates both explicit and implicit soliciting of data.

Specific-mode and any-mode are also used when data is solicited. In specific-mode, data is solicited from a particular terminal. In any-mode, data is solicited from all connected terminals. An application program might use these forms of solicitation in the following manner:

1. The application program initially solicits data from all of the terminals to which it has become connected.
2. The application program then issues a READ in any-mode, which is completed when one of the terminals responds to the solicitation.
3. The application program communicates with the terminal using WRITE and READ macro instructions issued in specific-mode. The READ macro instructions cause implicit solicitation.
4. When the transaction is completed, the application program issues a new SOLICIT macro instruction directed specifically at the terminal, so that a new READ issued in any-mode will be satisfied when the next transaction begins.

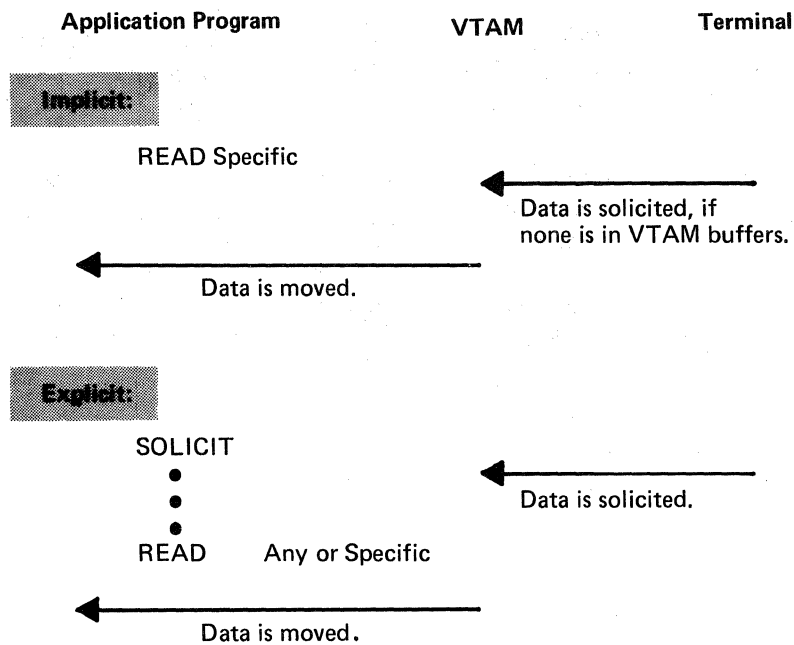


Figure 8-6. Implicit and Explicit Solicitation Using Basic Mode

When connection is established with a terminal in basic mode, the application program indicates the amount of data that each solicit request (implicit or explicit) is to obtain from that terminal. It is the application program's responsibility to determine when a new solicit request should be issued. The application program can designate that for each solicit request, VTAM:

Solicits only a *block* of data from the terminal. For start-stop terminals, a block ends with an EOB character; for BSC terminals, a block ends with an ETB or ETX character.

Solicits a *message* from the terminal. For start-stop terminals, a message ends in an EOT character. (Note that for start-stop terminals, a message is the same as a transmission.) For BSC terminals, a message ends with an ETX character. Messages consist of one or more blocks. Note that a message in basic mode is not the same as a message in record mode.

Solicits a *transmission* from the terminal. For both start-stop and BSC terminals, a transmission ends with an EOT character. Transmissions are comprised of one or more messages (for start-stop terminals, one or more blocks).

Solicits the terminal *continuously* until the application program cancels the solicitation.

**Soliciting Blocks:** When data is solicited a block at a time and an error occurs during transmission, the terminal needs to recover only a limited amount of data. However, since the application program must frequently reissue a solicit request (to acknowledge the previous block and obtain a new one), data throughput over the communication line is reduced. Block solicitation is appropriate when an unusually high number of line errors is expected and when the length of retransmitted data must be kept to a minimum, even at the expense of slower response times and poorer line utilization. The user must authorize the solicitation of blocks in the application program's APPL definition statement.

**Soliciting Messages and Transmissions:** The lengths of messages and transmissions are not as closely dependent on the type of terminal as are block lengths. Message and transmission lengths are usually established by the terminal's operator and the nature of the application. The lengths of messages and transmissions from a remote job entry station, for example, are determined by the number of cards in each job deck, and the number of job decks available for sending at one time.

Since messages and transmissions tend to be much longer than blocks, message and transmission solicitation requires recovery of more data when an I/O error is detected. However, with these forms of solicitation, data transfer can be more efficient, because the acknowledgments and resolicitations needed to obtain the blocks of data making up the message or transmission are performed by the communications controller not the application program.

Solicitation of messages and transmissions is appropriate for applications that require short response times but can tolerate lengthy transmissions when required.

The choice between message solicitation and transmission solicitation (which can be made only for BSC terminals) depends on how much time between data transfers is acceptable. With transmission solicitation, delays between data transfers are minimized, although more data must be recovered if errors occur.

**Continuous Solicitation:** The advantages and disadvantages of continuous solicitation are the opposite of those of block solicitation. By soliciting continuously, the application program can obtain data with the minimum of programming. However, the application program must determine when solicitation should cease, and must explicitly tell VTAM when to do so. If the solicitation must be interrupted frequently, the efficiency is lost.

Continuous solicitation is appropriate for batch input applications, where data transfers are relatively frequent and delays between blocks, messages, and transmissions must be minimized.

## Special I/O Operations

The application program can initiate the following I/O operations with one request:

- Copy a remote 3277 Display Station's buffer into the buffer of any printer or display station attached to the same cluster control unit (COPYLBM or COPYLBT operation)

- Send a positive response with leading graphic characters to a System/3 or System/370 CPU and then read the terminal's next block of data (WRTPRLG and READ operations); or send a negative response with leading graphic characters to one of these terminals and then reread the block of data (WRTNRLG and READ operations)

- Write data beginning with a block of heading characters to a System/3 or System/370 CPU (WRTHDR and WRITE operations)

- Write data to a terminal from separate output data areas (gather-write) or read from a terminal into separate input data areas (scatter-read)

To use these facilities, the application program builds a *LDO control block* (set of logical device orders). Each LDO indicates the specific type of I/O operation (such as COPYLBM or READBUF), the data area to be used, and an optional indicator that links the LDO to a following one. In both form and manner of use, LDOs resemble channel command word (CCW) programs. A set of LDOs is executed with a DO macro instruction. By using LDOs, the application program can request I/O operations that are not available with the conventional macro instructions like READ and WRITE.



## Special Processing Options

When connection is established with a terminal, the application program can designate rules that VTAM is to follow during subsequent communication with that terminal. The extent of solicitation described above—block, message, transmission, or continuous—is one example. Other options, most of which relate to NCP processing, can be selected by the application program (some options are not available for all types of terminals):

- VTAM can treat the receipt of leading graphic characters as either a normal condition or as an error condition. These options are called the LGIN and LGOUT options. (The names of these options, like those that follow, are the names coded as part of the PROC operand of the NIB macro instruction describing the terminal.)
- The application program can allow the communications controller to insert idle device-control characters into output data, or it can prevent the insertion of these characters (TMFLL option). If the communications controller is prepared to receive intermediate transmission blocks (ITBs) from a terminal, the application program can allow the communications controller to insert an error information byte (EIB) into each block, or it can prevent the insertion of EIBs (EIB option). The application program can use the EIBs to perform error recovery (retries) on a subblock basis, rather than on a block basis.
- The application program can override any text time-out limitation that the communications controller might otherwise use with the terminal (TIMEOUT option).
- The application program can prevent the communications controller from employing error recovery procedures if an error is detected during output to the terminal, during input from the terminal, or during either input or output (ERPIN and ERPOUT options).
- For some start-stop terminals, the application program can determine whether the communications controller is to monitor the terminal for attention interruptions and whether or not to notify the application program when the attention interruption is detected (MONITOR option). VTAM notifies the application program by scheduling its ATTN exit routine. (These are attention interruptions detected when the application program is not communicating with the terminal; attention interruptions that occur *during* an I/O operation are always brought to the attention of the application program by an RPL return code.)
- The application program can insert its own line-control characters into output data, or it can allow VTAM to do so (ELC option).
- The application program can send all data to the terminal in transparent text mode (BINARY option).

All of these options can be specified for each terminal. Unless the application program issues a request to change the rules, they remain in effect as long as the terminal is connected.

## Communicating with a BSC or Local 3270 Terminal in Record Mode

The BSC or local 3270 is not defined as a logical unit; however, the application program can communicate with it, in *record mode* as if it were an SNA 3270. The restrictions listed in Chapter 5 that apply to the SNA 3270 also apply to the BSC or local 3270 in record mode. Because of the more limited capabilities of the BSC or local 3270, the following additional restrictions also apply:

Bracket protocol must be used. If the application has no use for brackets, the first I/O operation of a connection and disconnection can be considered to be one bracket. Both the application program and the 3270 can begin a bracket. The first input from a 3270 that begins an NCP session is marked as the beginning of the bracket. All subsequent messages received from the 3270 during the NCP session indicate that the bracket is being continued. The 3270 cannot end a bracket; this can only be done by the application program.

SNA unformatted system services are not available to the BSC or local 3270 (this includes the ability to specify session parameters). The network solicitor must be used. (See “The Network Solicitor” earlier in this chapter for more information.)

**Note:** *The application program can communicate with the BSC or local 3270 in the same manner used to communicate with other BSC and start-stop terminals. See “Basic-Mode Concepts” earlier in this chapter.*



## APPENDIX A. SUPPORTED TERMINALS

This appendix lists the terminals and terminal features supported by VTAM. Terminals that are equivalent to those explicitly supported may also function satisfactorily. The user is responsible for establishing equivalency. IBM assumes no responsibility for the impact that any changes to the IBM-supplied products or programs may have on such terminals.

Where the terminal support list states that a terminal is “supported as” another terminal (for example, terminal *x* is “supported as” terminal *y*), it means that terminal *x* is defined to VTAM and uses VTAM facilities in the same manner as terminal *y*. This does not mean that the terminals have similar processing capabilities or physical characteristics. For example, a 3274 Model 1A is supported locally as a local 3791 controller. However, the data exchanged between an application program and the 3274 and the disposition of the data after it reaches the 3274 is not necessarily the same as for a 3791.

**Note:** For terminals on BSC and start-stop lines, VTAM receives and transmits data only in extended binary-coded-decimal interchange code (EBCDIC). The NCP translates EBCDIC messages from VTAM to the appropriate transmission codes for remote BSC and start-stop terminals. The NCP also translates all messages in transmission codes other than EBCDIC to EBCDIC before sending them to VTAM. For terminals on SDLC lines, VTAM receives and transmits data in EBCDIC or any other code. The NCP does no translation. The VTAM application program must do any translating from another code to EBCDIC and from EBCDIC to another code.

### SNA Terminals

SNA terminals consist of local and remote terminals. Local terminals are attached directly to the CPU on a channel. Remote terminals are attached on SDLC lines to either a local or a remote communications controller. The communications controller must contain a network control program.

An *SNA terminal product* is an IBM terminal product for which a PU (physical unit) statement and at least one LU (logical unit) statement are required when defining the network to VTAM.

**Note:** Features of an *SNA terminal product* and of the devices that are attached to it are a function of the particular *SNA terminal product*, not of VTAM.

#### Local SNA Terminals

VTAM supports the following SNA terminals attached on a channel.

#### 3270

3270 Information Display System. The following devices are supported:

3274 Control Unit (Model 1A), which is supported as a 3791. The following devices can be attached to the 3274:

3277 Display Station (Models 1 and 2)

3278 Display Station (Models 1 and 2)

3284 Printer (Models 1 and 2)

3286 Printer (Models 1 and 2)

3287 Printer (Models 1 and 2)

3288 Line Printer (Model 2)

3289 Line Printer (Models 1 and 2)

3790

3790 Communication System. The following devices are supported:

3791 Controller. The 3791 is required for attaching any of the devices listed below as part of the 3790 system:

3760 Dual Entry System

3277 Display Station

3288 Line Printer

3792 Auxiliary Control Unit. Support is included for the optional attachment of a 3793 Keyboard-Printer and a 2741 Communication Terminal.

3793 Keyboard-Printer

### **Remote SNA Terminals**

VTAM supports the following SNA terminals attached on an SDLC line to a local or remote communications controller.

*Note: SNA physical units (cluster controllers) cannot be operated in the same network containing start-stop or BSC terminals if that network is controlled by a 3704. Mixed operation of this type requires a 3705.*

3270

3270 Information Display System on nonswitched lines. The following devices in the 3270 system are supported in the VTAM network:

3271 Control Unit (Models 11 and 12). The 3271 allows attachment of the 3277, 3284 (Models 1 and 2), 3286, and 3288. Support is included for the following 3271 optional features:

1200—ASCII Code

1550—Copy

9761—EBCDIC Code

3274 Control Unit (Model 1C). The 3274 allows attachment of the 3277 (Models 1 and 2), 3278 (Models 1 and 2), 3284 (Models 1 and 2), 3286 (Models 1 and 2), 3287 (Models 1 and 2), 3288 (Model 2), and 3289 (Models 1 and 2). Support is included for the following 3274 optional feature:

9084—ASCII-B Character Set

3275 Display Station (Models 11 and 12). The 3275 is required for attaching the 3284 Model 3. Support is included for the following 3275 optional features:

1200—ASCII Code

6350—Selector Light Pen

9089—EBCDIC Character Set

9761—EBCDIC Code

3276 Control Unit Display Station (Models 11 and 12). The 3276 allows attachment of the 3278 (Models 1 and 2) and 3287 (Models 1 and 2).

3277 Display Station attached to a 3271 or 3274. Support is included for the following 3277 optional features:

6350—Selector Light Pen

9089—EBCDIC Character Set

3278 Display Station (Models 1 and 2), attached to a 3274 or 3276. Support is included for the following optional features:

6350—Selector Light Pen

9082—EBCDIC Character Set

3284 Printer (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3284 optional feature:

9089—EBCDIC Character Set

3284 Printer (Model 3). The 3284 Model 3 is attached to a 3275. Support is included for the following 3284 optional feature:

9089—EBCDIC Character Set

3286 Printer (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3286 optional feature:

9089—EBCDIC Character Set

3287 Printer (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3287 optional feature:

9082—EBCDIC Character Set

3288 Line Printer. When attached to a 3271 Model 12, the 3288 is supported as a 3286 Model 2. The 3288 Model 2 can also be attached to a 3274 or 3276. Support is included for the following 3288 optional feature:

9089—EBCDIC Character Set

3289 Line Printer (Models 1 and 2), attached to a 3274.

### 3600

3600 Finance Communication System on nonswitched lines. The following devices in the 3600 system are supported in the VTAM network:

3601 Finance Communication Controller. The 3601 is required for attaching the 3604, 3610, 3611, 3612, and 3618 as part of the 3600 system.

3604 Keyboard Display

3610 Document Printer

3611 Passbook Printer

3612 Passbook and Document Printer

3614 Consumer Transaction Facility. VTAM supports attachment of the 3614 to a 3704/3705 or a 3601. When attached to a 3601, a user-written 3601 application program is required.

3618 Administrative Line Printer

### 3650

3650 Retail Store System on switched or nonswitched lines. The following devices in the 3650 system are supported in the VTAM network:

3651 Store Controller (Models A50 and B50). The 3651 is required for attaching the 3653, 3657, 3275, and 3659 as part of the 3650 system.

3653 Point of Sale Terminal.

3657 Ticket Unit.

3275 Display Station (Model 3). Support is included for the optional attachment of a 3284 Printer (Model 3).

3659 Remote Communications Unit. The 3659 requires a 2400 bps nonswitched line.

### 3660

3660 Supermarket System on switched lines. The following devices in the 3660 system are supported in the VTAM network:

3651 Store Controller (Model 60). The 3651 is required for attaching the 3663 and 3669 as part of the 3660 system.

3663 Supermarket Terminal. VTAM supports the optional attachment of the 3666 Checkout Scanner to a 3663.

3669 Store Communications Unit.

**3767** 3767 Communication Terminal on switched or nonswitched lines. Support is included for the following 3767 optional features:

SDLC adapter, unless one of the start-stop features is specified  
1201—ASCII Code

**3770** 3770 Data Communication System on switched or nonswitched lines. The following devices in the 3770 system are supported by VTAM:

3771 Communication Terminal  
3773 Communication Terminal  
3774 Communication Terminal  
3775 Communication Terminal  
3776 Communication Terminal  
3777 Communication Terminal

VTAM requires the following features for all the devices in the 3770 system:

1460—SDLC/BSC, Switch Control, or  
1470—SDLC

Support is included for the following 3770 optional feature:

1201—ASCII Code

**System/32** System/32 Batch Work Station on switched or nonswitched lines supported as a 3770.

**3790** 3790 Communication System on switched or nonswitched lines. The following devices in the 3790 system are supported in the VTAM network:

3791 Controller. The 3791 is required for attaching the 3793, 3277, and 3792 as part of the 3790 system.  
3793 Keyboard-Printer.  
3277 Display Station.  
3792 Auxiliary Control Unit. Support is included for the optional attachment of a 3793 Keyboard-Printer and a 2741 Communication Terminal.

**5937** 5937-S01 Industrial Terminal on nonswitched lines. The 5937 is supported as a 3275.

## Non-SNA Terminals

Start-stop and BSC terminals can be attached to either a local or a remote communications controller in network control mode. Local 3270 terminals are attached by a channel to the CPU. Start-stop and BSC terminals (except for the 3270) are supported using basic mode macro instructions. The BSC 3270 and local 3270 terminals are supported using either the basic-mode or record-mode macro instructions.

## *Local Non-SNA Terminals*

### **Local 3270**

Application programs can use either basic-mode or record-mode macro instructions to communicate with local 3270 Information Display Systems. When basic-mode macro instructions are used, the local 3270 systems are viewed by VTAM as non-SNA systems. The following devices in the 3270 system are supported as local non-SNA terminals:

3272 Control Unit (Models 1 and 2). The 3272 allows attachment of the 3277 (Models 1 and 2), the 3284 (Models 1 and 2), 3286 (Models 1 and 2), and 3288 (Model 2).

3274 Control Unit (Model 1B), supported as a 3272. The 3274 allows attachment of the 3277 Display Station (Models 1 and 2), 3278 Display Station (Models 1 and 2), 3284 (Models 1 and 2), 3286 (Models 1 and 2), 3287 (Models 1 and 2), 3288 (Model 2), and 3289 (Models 1 and 2).

3277 Display Station (Models 1 and 2), attached to a 3272 or 3274. Support is included for the following 3277 optional features:

- 6350—Selector Light Pen
- 9089—EBCDIC Character Set

3278 Display Station (Models 1 and 2), attached to a 3274. Support is included for the following 3278 optional features:

- 6350—Selector Light Pen
- 9098—EBCDIC Character Set

3284 Printer (Models 1 and 2), attached to a 3272 or 3274. Support is included for the following 3284 optional feature:

- 9089—EBCDIC Character Set

3286 Printer (Models 1 and 2), attached to a 3272 or 3274. Support is included for the following 3286 optional feature:

- 9089—EBCDIC Character Set

3287 Printer (Models 1 and 2), attached to a 3274. Support is included for the following 3287 optional feature:

- 9082—EBCDIC Character Set

3288 Printer (Model 2). When attached to a 3272 Model 2, the 3288 is supported as a 3286 Model 2. The 3288 can also be attached to a 3274. Support is included for the following 3288 optional feature:

- 9089—EBCDIC Character Set

3289 Printer (Models 1 and 2), attached to a 3274.

Support is not included for the ASCII character set features for local 3270 terminals.

### **Start-Stop Terminals (Remote)**

Start-stop terminals can be attached to either a local or remote communications controller in network control mode. Application programs can communicate with start-stop terminals only through the basic mode of VTAM. The devices listed below are supported:

### **1050**

1050 Data Communication System on either switched or nonswitched lines. The following devices in the 1050 system are supported by VTAM:

1051 Control Unit (Models 1 and 2). The 1051 is required for attaching any of the devices listed below as part of the 1050 system. Support is included for the following 1051 optional features:



- 1313—Automatic EOB
- 2953—Open Line Detection
- 4795—Line Correction
- 4796—Line Correction Release
- 5465—Open Line Detection
- 6100—Receive Interrupt
- 9698—Text Time-Out Suppression
- 9700—Transmit Interrupt

1052 Printer-Keyboard (Models 1 and 2). Support is included for the following 1052 optional features:

- 1313—Automatic EOB
- 9567, 9597—PTTC/BCD Code
- 9571, 9591—PTTC/EBCD Code

1053 Printer (Model 1). Support is included for the following 1053 optional features:

- 9567, 9597—PTTC/BCD Code
- 9571, 9591—PTTC/EBCD Code

1054 Paper Tape Reader (Model 1).

1055 Paper Tape Punch (Model 1).

1056 Card Reader (Models 1 and 3).

1057 Card Punch (Model 1).

1058 Printing Card Punch (Models 1 and 2).

1092 Programmed Keyboard.

1093 Programmed Keyboard.

#### 2740 Model 1

2740 Communication Terminal (Model 1). Support is included for the following optional features for a 2740 on a switched line:

- 3255—Dial-up
- 8028—Transmit Control

Support is included for the following optional feature for a 2740 on a nonswitched line:

- 7479—Station Control

Support is included for the following optional features for a 2740 on either a switched or a nonswitched line:

- 6114—Record Checking
- 9567, 9597—PTTC/BCD Code
- 9571, 9591—PTTC/EBCD Code
- Correspondence Code

**Note:** VTAM does not support the 2760 Attachment feature (8301).

#### 2740 Model 2

2740 Communication Terminal (Model 2) on nonswitched lines. Support is included for the following 2740 optional features:

- 1495, 1496—Buffer Expansion
- 1499—Buffer Receive
- 6114—Record Checking
- 9571, 9591—PTTC/EBCD Code

- 2741** 2741 Communication Terminal (Model 1) on either switched or nonswitched lines. Support is included for the following 2741 optional features:
- 3255—Dial-up
  - 4708—Receive Interrupt
  - 7900—Transmit Interrupt
  - 9567, 9597—PTTC/BCD Code
  - 9571, 9591—PTTC/EBCD Code
  - Correspondence Code
- 3767** 3767 Communication Terminal (Models 1 and 2) supported as a 2740 Model 1, 2740 Model 2, or 2741.
- As a 2740 Model 1, on switched or nonswitched lines, VTAM requires:
- 7111—2740 Model 1 Start-Stop
- An optional feature is:
- 9560—Station Control
- As a 2740 Model 2, on nonswitched lines, VTAM requires:
- 7112-2740 Model 2 Start-Stop
- As a 2741, on switched or nonswitched lines, VTAM requires:
- 7113—2741 Start-Stop
- 5100 Portable Computer** 5100 Portable Computer as a 2741 on switched or nonswitched lines.
- Magnetic Card Typewriter** Communicating Magnetic Card SELECTRIC® Typewriter on switched lines. This device is supported as a 2741 terminal on a switched line with correspondence code. Thus, it is supported by VTAM as if it were a 2741 Communication Terminal equipped with a standard correspondence keyboard and element, the Dial-up feature, the Receive Interrupt feature, and the Transmit Interrupt feature.
- System/7** System/7 Processor Station on switched or nonswitched lines. System/7 is supported as a 2740 Model 1 with Checking. VTAM requires:
- 1610—Asynchronous Communication Control
- World Trade Telegraph** World Trade Telegraph Terminals on nonswitched lines. Support is included for the following World Trade Telegraph optional features:
- Paper Tape Reader
  - World Trade TTY ZSC3 Figures Protected Code
  - World Trade TTY ITA2 International Telegraph Alphabet 2
- AT&T 83B3** AT&T 83B3 Selective Calling Station on nonswitched lines. Support is included for the following AT&T 83B3 optional feature:
- Paper Tape Reader

**WU 115A**

WU Plan 115A Outstations on nonswitched lines. Support is included for the following WU 115A optional feature:

Paper Tape Reader

**CPT-TWX**

CPT-TWX Terminal (Models 33 and 35) on switched lines. Support is included for the following CPT-TWX optional features:

Data Interchange Code (8 level)  
Even Parity  
Forced Parity

***Binary Synchronous  
Communications  
(BSC) Terminals  
(Remote)***

BSC terminals can be attached to either a local or remote communications controller in network control mode. Application programs can communicate with BSC terminals, except for 3270s, only through the basic mode of VTAM. Either the basic mode or the record mode of VTAM can be used to communicate with 3270s. The devices listed below are supported.

**2770**

2770 Data Communication System on either switched or nonswitched lines. The following devices in the 2770 system are supported by VTAM:

2772 Multipurpose Control Unit. The 2772 is required to support any of the other devices listed below as part of the 2770 system. The following 2772 feature is required by VTAM:

5010--Multipoint Data Link Control

Support is included for the following 2772 optional features:

1340--Automatic Answering  
1490--Buffer Expansion (256 bytes)  
1491--Buffer Expansion Additional (512 bytes)  
1910--Conversational Mode  
3250--Display Format Control  
3650--EBCDIC Transparency  
3860--144 Character Print Line (1491 and 5558 on 2203 required)  
4610--Identification  
4690--Keyboard Correction  
5890--Horizontal Format Control  
6555--Space Compression/Expansion  
7705--Synchronous Clock  
7950--Trans/Rec Monitor Print  
9140--Extended Re-Entry  
9402--Line Termination 2-Wire  
9761--EBCDIC Code  
9762--ASCII Code  
9936--Immediate WACK

50 Magnetic Data Inscrber. VTAM does not provide any editing of input from the 50.

545 Output Punch (Models 3 and 4).

1017 Paper Tape Reader (Models 1 and 2).

1018 Paper Tape Punch (Model 1).

1053 Printer (Model 1).

1255 Magnetic Character Reader. VTAM does not support 1255 Stacker Select.

2203 Printer (Models A1 and A2).  
2213 Printer (Model 1 or 2).  
2265 Display Station (Model 2).  
2502 Card Reader (Models A1 and A2).  
5496 Data Recorder.

**2780**

2780 Data Transmission Terminal on either switched or nonswitched lines. Support is included for the following 2780 optional features:

1340—Automatic Answering  
1350—Automatic Turnaround  
3401—Dual Communication Interface  
5010—Multiple Record Transmission  
5020—Multipoint Line Control  
5820—120-Character Print Line  
5821-144-Character Print Line  
6400—Selective Character Set (USASCII)  
7850—Terminal Identification  
8030—EBCDIC Transparency  
9150—Extended Retry Transmission  
9761—ASCII Code  
9762—EBCDIC Code

*Note: VTAM does not support the six-bit Transcode feature (9760).*

**2980**

2980 General Banking Terminal System on nonswitched lines. The 2980 system is supported by VTAM in the U.S. only. The following devices in the 2980 system are supported by VTAM:

2972 Station Control Unit (Model 8—RPQ 858160—and Model 11—RPQ 858231). Support is included for the following 2972 optional features:

RPQ 835503—Buffer Expansion  
RPQ 858165, 858182—96-Character Buffer

2980 Teller Station (Model 1—RPQ 835504—and Model 4—RPQ 858147)

2980 Administrative Station (Model 2—RPQ 835505)

2971 Remote Control Unit (Model 3—RPQ 858144)

*Note: VTAM does not support the Batched Message Input feature for 2980 stations.*

**3270 (Remote)**

3270 Information Display System on nonswitched lines. The following devices in the 3270 system are supported by VTAM:

3271 Control Unit (Models 1 and 2). The 3271 allows attachment of the 3277, 3284 (Models 1 and 2), 3286 and 3288. Support is included for the following 3271 optional features:

- 1550—Copy
- 9761—EBCDIC Code

3274 Control Unit (Model 1C), which is supported as a 3271. The 3274 allows attachment of the 3277 (Models 1 and 2), 3278 (Models 1 and 2), 3284 (Models 1 and 2), 3286 (Models 1 and 2), 3287 (Models 1 and 2), 3288 (Model 2), and 3289 (Models 1 and 2). Support is included for the following 3274 optional features:

- 9082—EBCDIC Character Set
- 9084—ASCII-B Character Set

3276 Control Unit Display Station (Models 1 and 2), which is supported as a 3271. The 3276 allows attachment of the 3278 (Models 1 and 2) and 3287 (Models 1 and 2). Support is included for the following 3276 optional features:

- 9082—EBCDIC Character Set
- 9084—ASCII-B Character Set

3275 Display Station (Models 1 and 2). The 3275 is required for attaching a 3284 Model 3. Support is included for the following 3275 optional features:

- 6350—Selector Light Pen
- 9089—EBCDIC Character Set
- 9761—EBCDIC Code

3277 Display Station (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3277 optional features:

- 6350—Selector Light Pen
- 9089—EBCDIC Character Set

3278 Display Station (Models 1 and 2), attached to a 3274 or 3276. Support is included for the following 3278 optional features:

- 6350—Selector Light Pen
- 9082—EBCDIC Character Set

3284 Printer (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3284 optional feature:

- 9089—EBCDIC Character Set

3284 Printer (Model 3) The 3284 Model 3 is attached to a 3275. Support is included for the following 3284 optional feature:

- 9089—EBCDIC Character Set

3286 Printer (Models 1 and 2), attached to a 3271 or 3274. Support is included for the following 3286 optional feature:

- 9089—EBCDIC Character Set

3287 Printer (Models 1 and 2), attached to a 3274 or 3276. Support is included for the following 3287 optional feature:

- 9082—EBCDIC Character Set

3288 Line Printer (Model 2). When attached to a 3271 Model 2, the 3288 is supported as a 3286 Model 2. The 3288 Model 2 can also be attached to a 3274. Support is included for the following 3288 optional feature:

9089—EBCDIC Character Set

3289 Line Printer (Models 1 and 2), attached to a 3274.

### 3735

3735 Programmable Buffered Terminal on either switched or nonswitched lines. Support is included for the following 3735 optional features:

5010—Multipoint Data Link Control

9761—EBCDIC Code

9762—ASCII Code

The following devices are also supported for the 3735:

5496 Data Recorder (Model 1)

3286 Printer (Model 3)

### 3740

3740 Data Entry System on either switched or nonswitched lines. The 3740 is supported as a System/3. Thus, only 3740 functions equivalent to System/3 functions are supported by VTAM. The following devices in the 3740 system are supported by VTAM:

3741 Data Station (Model 2) and 3741 Programmable Work Station (Model 4) on switched or nonswitched lines. The 3741 is required for attaching the 129, 3713, 3715, and 3717. (The 3717 may be attached only to a 3741 Model 2.) Support is included for the following 3741 optional features:

1680—Expanded Communications

1685—Expanded Communications/Multipoint Data Link Control

5450—Operator Identification Card Reader

7850—Terminal Identification

129 Card Data Recorder (Model 2)

3713 Printer (Model 1)

3715 Printer (Models 1 and 2)

3717 Printer (Model 1)

3747 Data Converter (Model 1) on switched or nonswitched lines. Support is included for the following 3747 optional feature:

1660—Communications Adapter

### 3750

3750 Switching System on nonswitched lines. The 3750 system is supported by VTAM for World Trade only. The following device is supported:

3751 Control Unit. Support is included for the following 3751 required features:

1450—Binary Synchronous Communication Adapter, Type A, or

1451—Binary Synchronous Communication Adapter, Type B

**3770** 3770 Data Communication System supported as a 2770 on switched or nonswitched lines. The following devices in the 3770 system are supported by VTAM:

3771 Communication Terminal (Models 1 and 2) supported as a 2772.

3773 Communication Terminal (Models 1 and 2) supported as a 2772.

3774 Communication Terminal (Model 1) supported as a 2772.

3775 Communication Terminal (Model 1) supported as a 2772.

3776 Communication Terminal (Model 1) supported as a 2772 or 3780.

VTAM requires the following features for all the devices in the 3770 system:

1460—SDLC/BSC, Switch Control, or

1461—BSC Point-to-Point, or

1462—BSC Multipoint

Support is included for the following 3770 optional feature:

1201—ASCII Code

**3780** 3780 Data Communications Terminal on either switched or nonswitched lines. If the 3780 does not include the card-punch component, 3780 is supported as a 2770 without the Component Select feature. If the 3780 does include the card-punch component, the 3780 is supported as a 2770 with the Component Select feature. Thus only 3780 functions equivalent to 2770 functions are supported by VTAM. The following optional features are supported for the 3780:

3601—EBCDIC Transparency

5010—Multipoint Data Link Control

5701—Print Positions, Additional

9761—EBCDIC Code

9762—ASCII Code

*Note: Space Compression/Expansion is not supported for the 3780.*

**5275** 5275 Direct Numerical Control Station (Model 1) supported as a 3275 Model 1 or 3275 Model 2 with EBCDIC Code and EBCDIC Character Set on nonswitched lines. The 5275 is supported by VTAM in the U.S. only.

**5937** 5937-S01 Industrial Terminal on nonswitched lines. The 5937 is supported as a 3275.

**System/3** System/3 Central Processing Unit on either switched or nonswitched lines. Support is included for the following System/3 optional features:

1315—Autocall (U.S. only)

2074—Binary Synchronous Communications Adapter

7477—Station Selection

7850—EBCDIC Transparency

9060—EBCDIC Code

9061—ASCII Code

**System/7**

System/7 Processor Station on either switched or nonswitched lines. System/7 is supported as a System/3 with the Binary Synchronous Communications Adapter (2074). IPL of System/7 on a nonswitched point-to-point line is not supported.

**System/32**

System/32 Batch Work Station on either switched or nonswitched lines. System/32 is supported as a System/3 with the Binary Synchronous Communications Adapter (2074).

**System/370**

System/370 as a remote station on switched or nonswitched lines. The System/370 CPU must be attached at the remote location to a 2701, 2703, local communications controller in emulation or network control mode, or an Integrated Communications Adapter (ICA). Support is actually provided for the communications control unit as indicated in Figure B-2 in Appendix B. The following lists the supported optional features for each control unit:

2701 Data Adapter Unit

- 1302—Autocall (U.S. only)
- 1303—Autocall (U.S. only)
- 1314—Autocall (U.S. only)
- 1355—Dual Code (U.S. only)
- 3455—Dual Code (World Trade only)
- 3463-3465—Dual Communication Interface
- 8029—Transparency
- 9060—EBCDIC Code
- 9061—ASCII Code
- 9062—6-Bit Transcode

2702 Transmission Control Unit on local channel

- 1290—Autocall (U.S. only)
- 1319—Autopoll (U.S. only)

2703 Transmission Control Unit on local channel

- 1340—Autocall (U.S. only)
- 1341—Autocall (U.S. only)
- 7715—EBCDIC Code
- 7716—ASCII Code
- 7717—6-Bit Transcode
- 8055—2741 Break
- 9100—Transparency for ASCII

2715 Transmission Control Unit (Model 1 for 2790) on local channel

- 3801—Expanded Capability
- 4850—Local 2740 Adapter

2715 Transmission Control Unit (Model 2 for 2790) on local channel

- 3801—Expanded Capability
- 9401—Point-to-Point Non-Switched
- 9403—Multipoint Non-Switched

3704/3705 Communications Controller on local channel

- EBCDIC Code, ASCII Code, Autopoll (U.S. only), and EBCDIC Transparency do not have special feature codes in the 3704/3705
- EP/VS
- NCP/VS
- PEP
- 8002—Two-Channel Switch



ICA for the System/370 Model 125

4640 Integrated Communications Adapter

ICA for the System/370 Model 135

4640—Integrated Communications Adapter (EBCDIC standard feature)

9673-9680—Transparency

9681-9688—ASCII Code

9689-9696—6-Bit Transcode

## APPENDIX B. REMOTE STATION VERSUS REMOTE CONTROLLER

Because VTAM supports the communications controller both remotely attached and as a remote station, a distinction should be made between the connections of local and remote communications controllers. In a VTAM network, there are two ways in which two communications controllers can be connected to each other. One way is to have a remote communications controller attached as a satellite of a local controller. The remote connection is depicted in Figure B-1. The other way is to connect two independent, local communication controllers. The connection of local communications controllers is depicted in Figure B-2.

In Figure B-1, both VTAM and the NCP in the local communications controller control the remote communications controller. They recognize and use the remote unit as a communications controller, and they communicate with it, directing its attached devices. All control for the remote communications controller must emanate from the single host computer and be passed through the local communications controller.

Figure B-2 also represents a connection between two communications controllers, but in this case, neither controller is viewed as a remote communications controller by the other. Each controller with its attached host computer (containing independent VTAMs) is viewed by the other as a single terminal. In fact, there are two telecommunications networks almost totally independent of each other. In this type of connection, the two communications controllers treat each other as *remote stations*, not as remote communications controllers with the processing capability of an NCP.

In the network configuration depicted in Figure B-1, VTAM in CPU A can directly address any terminal attached to communications controller B. Figure B-2 depicts a network in which the VTAM in CPU A *cannot* directly address a terminal attached to communications controller B.

Tracing a message through the two systems helps to clarify the differences between the two types of attachments. Suppose an application program in CPU A is to send a message to terminal C:

*In the system depicted by Figure B-1:*

To send the message, the application program must be executing in the CPU with VTAM. The application program must request connection, using VTAM facilities, to terminal C. After the connection is completed, the application program must request that VTAM transmit the message to the terminal.

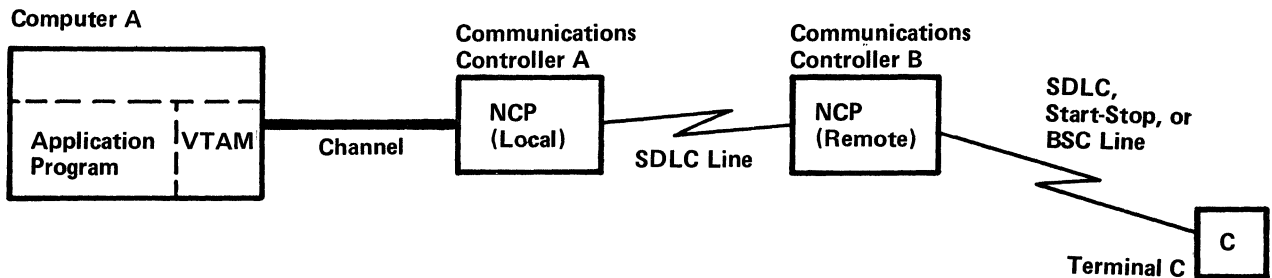


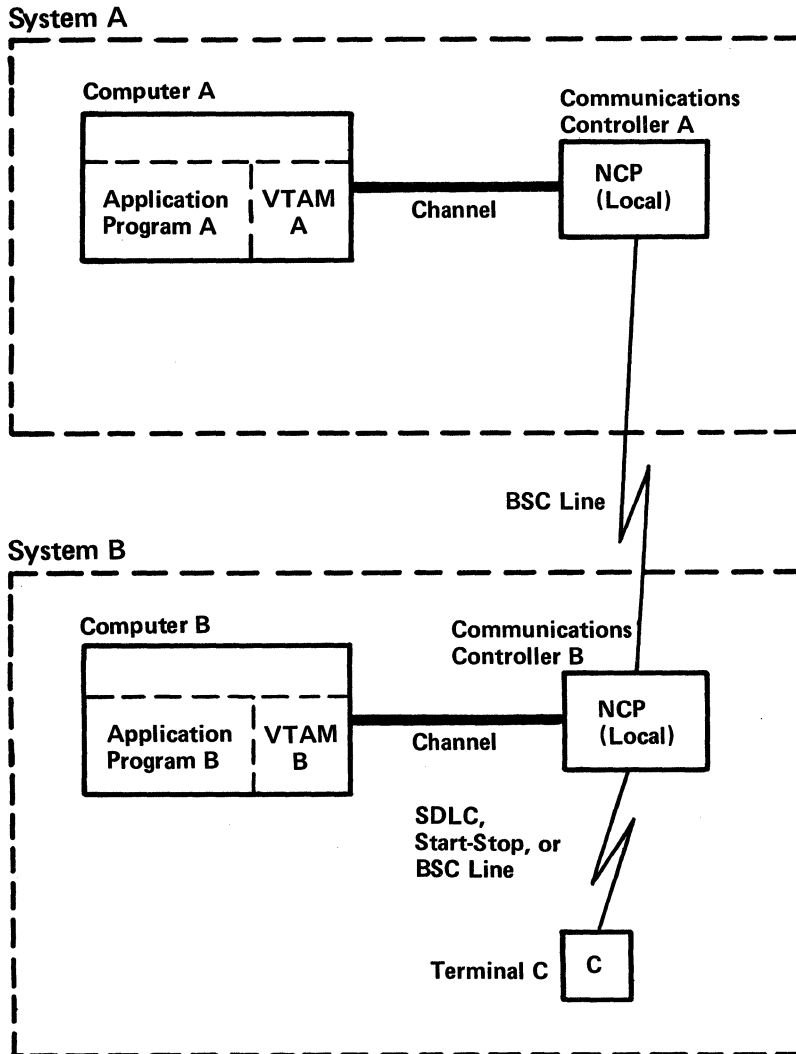
Figure B-1. A Remotely Attached Communications Controller

Upon receiving the request for data transmission, VTAM verifies that the terminal is in its network and transmits the message to the local communications controller. This controller (A in the figure), in turn, determines that the request is for a terminal attached to the remote unit. Communications controller A then transmits the message to communications controller B. The remote controller routes the message to the terminal.

Note that the application program need not be aware of how the terminal is attached. For example, if the terminal is a 3270, it could be attached to CPU A, to communications controller A, or to communications controller B. Regardless of the attachment, the application program uses the same procedure to connect and communicate with the terminal.

*In the system depicted by Figure B-2:*

To send a message from an application program executing in the CPU in system A to a terminal (terminal C) in system B, the application program must be aware that terminal C is in another system. Also a companion application program must be executing in system B. The two application programs must be designed to work in coordination with each other.



*Note:* VTAM does not have to be the access method in both computers to use a controller as a remote station. For example, if another telecommunication access method were in Computer B, System B would still remain a remote station for VTAM in A

Figure B-2. Communications Controllers Attached as Part of Remote Stations

Application program A first requests VTAM for connection with the terminal that is system B. (Remember that each system is defined to the other as a single terminal.) The application program in CPU B also requests connection, but these requests are directed to VTAM B and are for terminal C and for the terminal that is system A.

To transmit the message, application program A requests (using basic mode macro instructions) that VTAM A transmit the message to the connected terminal (system B). VTAM A transmits the message to communications controller A which, in turn, writes the message to communications controller B. At this point, application program B must have issued a basic mode request to VTAM B for input from the terminal defined as system A. Upon receiving the message from controller A, communications controller B transmits it to VTAM B. VTAM B then sends the message to application program B. Using user-defined procedures, application program B determines that the message is destined for terminal C. So using VTAM B and communications controller B, application program B causes the message to be written to terminal C.

Thus when a communications controller is remotely attached (by a duplex line), its facilities and attached terminals are available directly to a single operating system. When two local communications controllers are attached to each other, two operating systems are employed. Each operating system has direct access only to the facilities and attached terminals of its communication controllers.



## APPENDIX C. SUMMARY OF MESSAGE CONTROL INFORMATION

This appendix provides a summary of indicators and commands that can be exchanged between an application program using VTAM and a logical unit. The indicators and commands in this appendix are those that can be included as a message or part of a message.

A message can include data and/or indicators and commands. A response can include:

The type of response, that is, a definite response 1 or a definite response 2 or both and positive or negative.

An explanation of the response if it is negative.

Change-direction and bracket indicators. (These indicators can also be sent on messages and are described below.)

The remainder of this appendix is a table of the message commands and indicators. The table is divided into columns; each column is defined as follows:

**Type of Indicator or Command:** Specifies the name of the indicator or command. If it has an abbreviation, the abbreviation is in parentheses.

**Function:** Describes the use of the indicator or command.

**VTAM Application Program Can Send/Receive:** Indicates whether the indicator or command can be transmitted or received by the application program.

**Macro Used or RPL Field Set:** Indicates the VTAM macro instruction or RPL field used in specifying the indicator or command.

**Data-Flow Type:** Indicates whether the indicator or command is sent in normal flow (DFSYN) or expedited flow (DFASY).

**Next Action Expected:** Indicates what action should occur following the exchange of indicator or command.

This table contains the following types of indicators and commands:

Normal-flow

Expedited-flow

Change-direction

Bracket

SESSIONC

Type of Command or Indicator	Function	Application Program Can Send/Receive	Macro Used or RPL Field Set	Data-Flow Type	Next Action Expected
------------------------------	----------	--------------------------------------	-----------------------------	----------------	----------------------

**Normal-Flow Commands**

Bid	Bid to begin a bracket	Send only	SEND with CONTROL=BID	DFSYN	Receive response that says whether or not a bracket can be started
Cancel	Tell receiver to purge chain elements already received	Send	SEND with CONTROL=CANCEL	DFSYN	Response indicates cancellation
		Receive	CANCEL set in CONTROL field	DFSYN	Purge chain elements
Chase	Determine that the receiver has no more responses to send	Send	SEND with CONTROL=CHASE	DFSYN	When a response to the CHASE is received, all responses are accounted for
		Receive	CHASE set in CONTROL field	DFSYN	Send a positive or exception response
Logical Unit Status (LUS)	Inform receiver of an unexpected condition	Send	SEND with CONTROL=LUS	DFSYN	Varies with condition
		Receive	LUS set in CONTROL field	DFSYN	Varies with condition
Quiesce Complete (QC)	Tell the receiver that the sender is now quiesced and will not send until released	Send	SEND with CONTROL=QC	DFSYN	Await Release-Quiesce command
		Receive	QC set in CONTROL field	DFSYN	Send to quiesced receiver
Ready to Receive (RTR)	Tell the VTAM application program that a bracket has ended and a request to begin one can be sent	Receive only	RTR set in CONTROL field	DFSYN	Send a message that includes BRACKET=BB

**Expedited-Flow Commands**

Quiesce-at-End-of-Chain (QEC)	Tell the receiver to quit sending now or, if chaining, at the end of this chain	Send	SEND with CONTROL=QEC	DFASY	Receiver should send a Quiesce-Complete command
		Receive	QEC set in the CONTROL field	DFASY	Finish sending, then send Quiesce-Complete command
Release Quiesce (RELQ)	Permit the receiver to send now	Send	SEND with CONTROL=RELQ	DFASY	Expect message from receiver
		Receive	RELQ set in CONTROL field	DFASY	Send a message, if desired

Type of Command or Indicator	Function	Application Program Can Send/Receive	Macro Used or RPL Field Set	Data-Flow Type	Next Action Expected
------------------------------	----------	--------------------------------------	-----------------------------	----------------	----------------------

**Expedited-Flow Commands (Cont.)**

Request-Shutdown (RSHUTD)	Request the VTAM application program to send a SHUTD or to disconnect the logical unit	Receive only	RSHUTD set in CONTROL field	DFASY	Send a SHUTD or disconnect
Shutdown-Complete (SHUTC)	Tel the receiver that preparation for shutdown is complete	Receive only	SHUTC set in CONTROL field	DFASY	Disconnect the logical unit
Shutdown (SHUTD)	Tell a logical unit to prepare to shut down	Send only	SEND with CONTROL=SHUTD	DFASY	The logical unit should send a SHUTC
Signal	Pass a 4-byte message with an agreed-upon meaning	Receive	SIGNAL set in in CONTROL field: values in SIGNAL and SIGDATA fields	DFASY	Installation-or terminal product-defined
		Send		DFASY	Installation-defined

**Change-Direction-Indicators**

The Change-Direction-Command indicator can be sent in a message that contains data, or with a bracket indicator. It can also be sent in a response which, can include a bracket indicator.

The Change-Direction-Request indicator can be sent in a message that contains a QEC, RELQ, or Signal command, or a bracket indicator. It can also be sent in a response which, can include a bracket indicator.

Change-Direction-Command (CMD)	Tell the receiver that it is now its turn to send	Send	SEND with CHNGDIR=CMD	DFSYN	Receive from the logical unit
		Receive	CMD in CHNGDIR field	DFSYN	Send to the logical unit
Change-Direction-Request (REQ)	Request the receiver to send a Chang-Direction-Command indicator	Send	SEND with CHNGDIR=REQ	DFASY or RESP	Check incoming messages or responses for CMD in CHNGDIR field
		Receive	REQ in CHNGDIR field	DFASY or RESP	Send CMD when able



Type of Command or Indicator	Function	Application Program Can Send/Receive	Macro Used or RPL Field Set	Data-Flow Type	Next Action Expected
------------------------------	----------	--------------------------------------	-----------------------------	----------------	----------------------

#### Bracket Indicators

The normal-flow commands, Bid and Ready-to-Receive, are used by the VTAM application program to determine whether it can send a Begin-Bracket indicator.

The bracket indicators can be sent in a message that contains data or a normal-flow command (except Bid or Ready-to-Receive). A Change-Direction indicator can also be sent in the same message or response.

Begin-Bracket (BB)	Begin a bracket	Send	SEND with BRACKET=BB	DFSYN	Receive response that indicates a bracket was begun
		Receive	BB set in BRACKET field	DFSYN	None
End-Bracket (EB)	End a bracket	Send	SEND with BRACKET=EB	DFSYN	Receive response that indicates the bracket was ended
		Receive	EB set in BRACKET field	DFSYN	Can mean "end-of-transaction"

#### SESSIONC Commands

These commands control and are sent separately from normal- and expedited-flow messages and their responses.

Bracket and change direction indicators cannot be sent with SESSIONC command messages or responses.

Clear	Stop all sending of normal-flow or expedited-flow messages and responses, clear pending ones from the network, and reset sequence numbers to 0	Send only	SESSIONC with CONTROL=CLEAR	Not Applicable	Reset sequence number to other than 0 (Optional) and restart message flow
Request-Recovery (RQR)	Tell the VTAM application program that a recovery procedure is required	Receive only	RQR set in CONTROL field in SCIP exit-routine	Not Applicable	Clear and reset sequence numbers to other than 0 (Optional) and restart message flow
Start-Data-Traffic (SDT)	Allow the sending of normal-flow or expedited-flow messages and responses	Send only	SESSIONC with CONTROL=SDT	Not Applicable	Normal message flow can resume
Set-and-Test Sequence-Number (STSN)	Exchange information with the logical unit so that sequence numbers can be determined and/or reset	Send only	SESSIONC with CONTROL=STSN and settings in IBSQAC and/or OBSQAC fields and IBSQVAL and/or OBSQVAL fields	Not Applicable	Receive response indicating logical unit reaction to STSN message

## APPENDIX D. COMPARIBILITY CONSIDERATIONS FOR BSC AND LOCAL 3270 TERMINALS AND SNA 3270 TERMINALS

Since SNA unformatted system services are unavailable to BSC and local 3270 terminals, problems may arise for installations that are currently using these terminals in record mode and that plan to replace them or use them together with SNA 3270 terminals.

The following guidelines may ease the transition from one type of terminal to the other. They should also permit both types to be used together without any apparent differences to the terminal operator or application program.

### *Changes Necessary when Replacing Terminals*

**NCP Definition:** PU and LU macro definition statements for the SNA 3270 must replace the CLUSTER and TERMINAL macros for the BSC 3270.

**Interpret Tables:** Interpret tables like those used for SNA terminals may be required for SNA 3270 terminals. Only the application-program name field (from the APPLID in an SNA unformatted system services logon) is interpreted for the SNA 3270. For BSC and local 3270 terminals, the entire logon character string is interpreted.

### *Logon Compatibility*

If the user permits a terminal operator to issue logons from a BSC or local 3270, the logon character string that is defined in its interpret tables or within a user-written network solicitor should conform to the syntax allowed by unformatted system services (as defined by a user-written USS definition table). If this is done, the logon sequence entered through a BSC 3270 will also work through an SNA 3270. For example:

```
LOGON APPLID(application)
LOGON application
LON
LOGON APPLID(application) LOGMODE(mode)
```

In addition, if the application program is to be unable to distinguish between a BSC or local 3270 and an SNA 3270, it should not examine logons (using INQUIRE LOGONMSG or INQUIRE SESSPARM commands). This restriction is necessary since the entire logon sequence is passed from a BSC or local 3270, whereas only the logon data (DATA field) is passed from an SNA 3270.

As an alternative to not examining the logon data, the user can write a network solicitor that truncates the logon character string and passes only the data portion of the logon to the application program. Using this modified network solicitor, the following types of logons could be entered from a BSC or local 3270 and handled by the application program as if they had been entered by an SNA 3270:

```
LOGON APPLID(application) DATA(user-data)
LOGON application,user-data
```

### **Using an Interpret Table for SNA Terminals**

The user can define an interpret table for SNA devices. This will convert logons entered through SNA devices into a form acceptable to non-SNA devices. This facility is not recommended because it precludes the use of the full range of SNA functions by the SNA terminal.

## ***Application Program Considerations***

Debugged application programs that have been written using record mode for BSC and local 3270 terminals can operate with SNA 3270 terminals, with the following exceptions:

- The program cannot examine logons unless modified to handle the SNA format as discussed above.
- Since the SNA 3270 sends segmented request units, the LOSTERM exit routine may be scheduled with a return code of 28 when segmenting errors are detected. The application program should be modified to handle this condition.

Application programs that use both the BSC and local 3270 and the SNA 3270 must be written according to the following rules:

- There should be logon compatibility as discussed above.
- Record mode must be used since SNA 3270 terminals cannot use basic mode.
- The restrictions listed in "Communicating with the 3270 Information Display System" in Chapter 5 must be carefully adhered to. Since VTAM does not enforce these restrictions for the SNA 3270, failure to adhere to these restrictions will cause unpredictable results.
- SNA services (logon mode, variable session parameters, 3270 non-bracket operation, conditional logoff) cannot be used; they are available to the SNA 3270 only.
- When an error is encountered on a SEND macro instruction that specifies End Bracket, the bracket is ended for local and SNA 3270 terminals, but not for BSC 3270 terminals. When retrying the SEND macro instruction, the Begin Bracket indicator can be specified. If a bracket error occurs because the bracket was not ended (for the BSC 3270), the SEND can be reissued after the bracket indicator is changed to End Bracket.

## GLOSSARY

This glossary defines terms and abbreviations that are important in this book. It does not include terms previously established for IBM operating systems and IBM products used with VTAM. Additional terms can be found by referring to the index, to prerequisite and corequisite books, and to the *IBM Data Processing Glossary*, GC20-1699.

IBM is grateful to the American National Standards Institute (ANSI) for permission to reprint its definitions from the *American National Standard Vocabulary for Information Processing* (Copyright© 1970 by American National Standards Institute, Incorporated), which was prepared by Subcommittee X3K5 on Terminology and Glossary of the American National Standards Committee X3. A complete commentary taken from ANSI is identified by an asterisk that appears between the term and the beginning of the commentary; a definition taken from ANSI is identified by an asterisk after the item number for that definition.

The symbol *ISO* at the beginning of a definition indicates that it has been discussed and agreed upon at meetings of the International Organization for Standardization Technical Committee 97/Subcommittee 1 (Data Processing), and has also been approved by ANSI.

The symbol *SCI* at the beginning of a definition indicates that it is reprinted from an early working document of ISO Technical Committee 97/Subcommittee 1 and that final agreement has not yet been reached among its participating members.

### A

**ACB.** Access method control block.

**accept.** In VTAM, to connect a terminal to a VTAM application program as the result of a logon. The logon may be originated by the terminal, the network operator, another application program, or VTAM. Contrast with *acquire*.

**access method control block (ACB).** A control block that links an application program to VSAM or VTAM.

**accounting exit routine.** In VTAM, an optional, user-written routine that collects statistics about connections and disconnections in the communication network.

**acquire.** In VTAM, to connect a terminal to a VTAM application program in the absence of a logon. The connection occurs at the program's initiative. Contrast with *accept*.

**active.** Pertaining to major node that is known to VTAM and is available for use or pertaining to a minor node that is connected to, or available for connection to, a VTAM application program. Contrast with *inactive*.

**any-mode.** In VTAM: (1) The form of a read or receive request that obtains data from one unspecified terminal. (2) The form of solicit request that solicits data from all eligible connected terminals. (3) The form of connection request that connects one unspecified terminal that has logged on. (4) Contrast with *specific-mode*. See also *continue-any mode*.

**application program identification.** The symbolic name by which a VTAM application program is identified to VTAM and other components of the network.

**application program major node.** In VTAM, a member (OS/VS) or book (DOS/VS) of the VTAM definition library that contains one or more APPL statements, each representing a VTAM application program.

**APPLID routine.** Synonym for *logon-interpret routine*.

**asynchronous operation.** In VTAM, an operation such as connection or data transfer in which the VTAM application program is allowed to continue execution while VTAM performs the operation. VTAM interrupts the program after the operation is completed.

**asynchronous request.** In VTAM, a request for an asynchronous operation.

**authorization exit routine.** In VTAM, an optional, user-written routine that approves or disapproves requests for connection and disconnection.

**authorized path.** In OS/VS2 VTAM, a facility that enables an authorized VTAM application program to specify that a data transfer or related operation be carried out in a faster manner than usual.

**automatic logon.** In VTAM, a logon performed by VTAM on behalf of a terminal and a VTAM application program without either of them having to request it. The capability to log on automatically is specified during VTAM definition and can be modified by the network operator.

**available.** In VTAM: (1) Pertaining to a terminal that is active, is not connected to an application program, and for which there is no pending logon. (2) Pertaining to an exit routine that has been specified by an application program and that is not being executed.

### B

**basic mode.** In VTAM, a mode of data transfer in which the VTAM application program can communicate with local 3270 Information Display Systems, start-stop terminals, and BSC terminals. Contrast with *record mode*.

**block.** In the basic mode of VTAM, a unit of data that is transmitted between a VTAM application program and a terminal.

**bracket.** In VTAM, an uninterruptible unit of work, consisting of one or more chains of request units and their responses, exchanged between an application program and a terminal. Examples are data base inquires/responses, update transactions, remote job entry output sequences to work stations, and similar applications.

**bracket protocol.** In SNA, a data flow control protocol in which exchanges between logical units (LUs) are achieved through the use of brackets, with one LU designated at session initiation as

the first speaker, and the other LU as the bidder. The bracket protocol involves bracket initiation and termination rules.

## C

**change-direction protocol.** In VTAM, a method of communication in which the sender stops sending on its own initiative, signals this fact to the receiver, and prepares to receive.

**character-coded.** In VTAM, pertaining to a logon or logoff command usually entered by a terminal operator from a keyboard and sent by a logical unit in character (unformatted) form. Contrast with *field-formatted*.

**CID.** Communication identifier.

**closedown.** The deactivation of a device, program, or system. See also *quick closedown*, *orderly closedown*.

**cluster controller.** See *cluster control unit* and *SDLC cluster controller*.

**cluster control unit.** A device that can control the input/output operations of more than one device. A remote cluster control unit can be attached to a host computer only through a communications controller. A cluster control unit 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 2972 Station Control Unit. See also *communications controller* and *SDLC cluster controller*.

**command.** (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, a request unit initiating an action or beginning a protocol; it is used in contrast with reply, which is a request unit (not a response) that is sent in reaction to a command. For example: Quiesce (a data flow control request), which is a command, while Quiesce Complete is the reply.

**communication control character.** \*A control character intended to control or facilitate transmission of data over communication networks.

**communication identifier (CID).** In VTAM, a key for locating the control blocks that represent an active session. The key, which consists of a pair of network addresses, is created when the session begins and deleted when the session ends.

**communication line.** Any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communication control unit, or connects one communication control unit with another.

**communications controller.** A type of communication control unit whose operations are controlled by a program stored and executed in the unit. Examples are the IBM 3704 and 3705 Communications Controllers.

**configuration restart.** In VTAM, the facility for recovering after a failure in the NCP or communications controller or after a loss of contact with a physical unit. Recovery may include reloading the NCP or restoring the network by means of a checkpoint. Restarting by means of a checkpoint requires the user to specify a VSAM data set in which VTAM records configuration data.

**connection.** In VTAM, the linking of VTAM control blocks in such a way that a VTAM application program can communicate with a terminal. Connection includes establishing and preparing the network path between the program and the terminal. See also *queued for connection*.

**continue-any mode.** In VTAM, a state into which a terminal is placed that allows its input to satisfy an input request issued in

any-mode. While this state exists, input from the terminal can also satisfy input requests issued in specific-mode. Contrast with *continue-specific mode*.

**continue-specific mode.** In VTAM, a state into which a terminal is placed that allows its input to satisfy only input requests issued in specific-mode.

**conversational write operation.** In the basic mode of VTAM, an operation wherein data is first sent to a terminal and data is then read from that terminal.

**converted command.** An intermediate form of a character-coded logon or logoff command produced by VTAM through use of an unformatted system services definition table. The format of a converted logon or logoff command is fixed; the unformatted system services definition table must be constructed so that the character-coded command (as entered by a *logical unit*) is converted into the predefined, converted command format. See also *character-coded*.

## D

**data flow.** In SNA, any of four flows in a given session, characterized as either primary-to-secondary or secondary-to-primary, and either normal or expedited.

**data transfer.** In data communication, the sending of data from one point in a communication network and the receiving of the data at another point in the network.

**data transmission.** The sending of data from one point in a communication network for reception elsewhere by means of a channel or communication line.

**definite response.** In SNA, a form of response requested in the request header for a request unit; the receiver is requested to return a response whether positive or negative. Contrast with *exception response*.

**definition statement.** In VTAM, the means of describing an element of the telecommunication system.

**device control character.** (ISO) A control character used for the control of ancillary devices associated with a data processing system or data communication system, for example, for switching such devices on or off.

**disconnection.** In VTAM, the dissociation of VTAM control blocks in such a way as to end a session between a VTAM application program and a connected terminal. The disconnection process includes suspending the use of the network path between the program and the terminal.

## E

**emulation mode.** A function of the network control program that enables a 3704 or 3705 Communications Controller to perform activities equivalent to those performed by an IBM 2701 Data Adapter Unit or an IBM 2702 or 2703 Transmission Control Unit. See also *network control mode*.

**exception message.** In communicating with a logical unit, a message that indicates an unusual condition such as a sequence number being skipped. When VTAM detects such a condition, it notifies the VTAM application program. VTAM and/or the VTAM application program provides sense information which is included in the response that is sent to the logical unit.

**exception response.** (1) In SNA, a response requested in the RH for a request unit; the receiver is requested to return a response only if it is negative. Contrast with *definite response*. (2) Synonym for *negative response*.

**exit list (EXLST).** In VSAM or VTAM, a control block that contains the addresses of user-written routines that receive control when specified events occur during execution; for example, routines that process logons or I/O errors.

**exit routine.** In VTAM, any of several types of special-purpose user-written routines. See *accounting exit routine*, *authorization exit routine*, *EXLST exit routine*, *logon-interpret routine*, and *RPL exit routine*.

**EXLST exit routine.** In VTAM, a user-written routine whose address has been placed in an exit list (EXLST) control block. See also *RPL exit routine*.

**expedited flow.** In SNA, a data flow that is independent of and controls the normal flow. Data flow is split into normal and expedited flows. Requests and responses on a given (normal or expedited) flow are processed sequentially within the path, but the expedited flow traffic may be moved ahead of the normal flow traffic within the path. Contrast with *normal flow*.

## F

**field-formatted.** In VTAM, pertaining to a logon or logoff command that is encoded into fields, each having a specified format such as binary codes, bit-significant flags, and symbolic names. Contrast with *character-coded*.

## H

**host computer.** (1) The primary or controlling computer in a multiple computer operation. (2) A computer used to prepare programs for use on another computer or on another data processing system; for example, a computer used to compile, link-edit, or test programs to be used on another system. (3) The primary or controlling computer in a data communication system. (4) In a VTAM telecommunication system, the processing unit in which VTAM resides.

**host system.** (1) A data processing system that is used to prepare programs and the operating environments for use on another computer or controller. (2) The data-processing system to which a communication system is connected and with which the system can communicate.

## I

**inactive.** In VTAM, pertaining to a major node that is unknown to VTAM and is unavailable for use, or pertaining to a minor node that is not connected to nor available for connection to a VTAM application program. Contrast with *active*.

**interpret table.** In VTAM, a user-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.

## L

**LDO.** Logical device order.

**line.** See *communication line*.

**line control.** The scheme of operating procedures and control signals by which a telecommunication system is controlled.

**line group.** One or more communication lines of the same type that can be activated and deactivated as a unit.

**local.** (1) Pertaining to the attachment of devices directly by I/O channels to a host computer. Contrast with *remote*. (2) In data communication, pertaining to devices that are attached to a controlling unit by cables, rather than by data links.

**local SNA major node.** In VTAM, a major node whose minor nodes are locally attached physical and logical units of one or more 3790 Communication Systems.

**local 3270 major node.** In VTAM, a major node whose minor nodes are locally attached 3270 terminals.

**logical device order (LDO).** In VTAM, a set of parameters that specify a data-transfer or data-control operation to local 3270 Information Display Systems and certain kinds of start/stop or BSC terminals.

**logical error.** In VTAM, an error condition that results from an invalid request; a program logic error.

**logical unit.** In SNA, one of three types of network addressable units (NAUs). It is the port through which an end user accesses function management in order to communicate with another end user. It is also the port through which the end user accesses the services provided by the system services control point (SSCP). It must be capable of supporting at least two sessions – one with the SSCP, and one with another logical unit. It may be capable of supporting many sessions with other logical units. See also *physical unit*, *system services control point*.

**log off.** In VTAM, to request that a terminal be disconnected from a VTAM application program.

**logoff.** In VTAM, a request that a terminal be disconnected from a VTAM application program.

**log on.** In VTAM, to request that a terminal be connected to a VTAM application program.

**logon.** In VTAM, a request that a terminal be connected to a VTAM application program. See also *automatic logon* and *simulated logon*.

**logon data.** In VTAM: (1) The data portion of a field-formatted or character-coded logon from an SNA terminal. (2) The entire logon sequence from a non-SNA terminal. (3) Synonymous with *logon message*.

**logon-interpret routine.** In VTAM, a user-written routine that translates logon data. It may also verify the logon. Synonymous with *APPLID routine*.

**logon message.** Synonym for *logon data*.

**logon mode.** In VTAM, the communication protocols that govern a session between a logical unit and a VTAM application program. Synonymous with *session parameters*.

**logon mode name.** In VTAM, the symbolic representation used by a VTAM application program and a logical unit to refer to a logon mode.

**logon mode table.** In VTAM, a set of macro-generated constants making up one or more logon modes. Each logon mode is associated with a logon mode name.

## M

**major node.** In VTAM, a set of minor nodes that can be activated and deactivated as a group. See also *minor node*.

**message.** (1) \*An arbitrary amount of information whose beginning and end are defined or implied. (2) For BSC devices, the data unit from the beginning of a transmission to the first ETX character, or between two ETX characters. For start/stop devices "message" and "transmission" have the same meaning. (3) (SC1) A sequence of characters used to convey data. The sequence usually consists of three parts: the heading, the text,

and one or more characters used for control or error-detection purposes. (4) In telecommunications, a combination of characters and symbols transmitted from one point to another.

**minor node.** In VTAM, an element of the telecommunication network that can be activated or deactivated by the VARY command. See also *major node*.

**MTA.** Multiple terminal access.

**multiple terminal access (MTA).** A feature of the network control program that permits it to communicate with a variety of dissimilar, commonly used start-stop terminals over the same switched network connection.

**multithread application program.** A VTAM application program that processes many requests from many terminals concurrently. Contrast with *single-thread application program*.

## N

**NCP.** Network control program.

**NCP major node.** In VTAM, a major node whose minor nodes are defined through NCP generation.

**negative response.** A response indicating that a message did not arrive successfully or is unacceptable. Synonymous with *exception response*. Contrast with *positive response*.

**negative response to polling limit.** For a start-stop or BSC terminal, the maximum number of consecutive negative responses to polling that the communications controller accepts before suspending polling operations.

**network.** (1) (SC1) The assembly of equipment through which connections are made between terminal installations. (2) In data communication, a configuration in which two or more terminal installations are connected.

**network control mode.** The functions of a network control program that enable it to direct a communication controller to perform telecommunication activities such as polling, device addressing, dialing, and answering. See also *emulation mode*.

**network control program (NCP).** A program, generated by the user from a library of IBM-supplied modules, that controls the operation of the communications controller.

**network control program generation.** The process, performed in a host system, of assembling and link-editing a macro instruction program to produce a network control program.

**network definition.** In VTAM, the process of defining the identities and characteristics of each node in the telecommunication system and the arrangement of the nodes in that system.

**network operator.** (1) The person responsible for controlling the operation of a telecommunication network. (2) A VTAM application program authorized to issue network operator commands.

**network operator command.** A command used to monitor or control the telecommunication network.

**network operator console.** A system console or terminal in the network from which a network operator controls a telecommunication network.

**network operator logon.** A logon requested on behalf of a terminal by means of a network operator command.

**NIB.** Node initialization block.

**NIB list.** A series of contiguous node initialization blocks.

**node.** (1) An addressable point in a data communication network. (2) In VTAM, a point in a telecommunication system defined by a symbolic name. See also *major node* and *minor node*.

**node initialization block (NIB).** In VTAM, a control block associated with a particular terminal that contains information used by the VTAM application program to identify the terminal and indicate how communication requests directed at the terminal are to be processed.

**node name.** In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

**non-SNA terminal.** In VTAM, a terminal that is part of a local 3270 Information Display System or a terminal supported by VTAM that uses start-stop or BSC protocol.

**normal flow.** In SNA, a data flow that is used for most requests and responses. Data flow is split into normal and expedited flows. The expedited flow is independent of and used to control the normal flow. Requests and responses on a given (normal or expedited) flow are processed sequentially within the path, but the expedited flow traffic may be moved ahead of the normal flow traffic within the path. Contrast with *expedited flow*.

## O

**orderly shutdown.** The orderly deactivation of VTAM and the telecommunication network. An orderly shutdown does not take effect until all application programs have been disconnected from VTAM. Until then, all data transfer operations continue. Contrast with *quick shutdown*.

## P

**partitioned emulation programming (PEP).** A feature of the network control program, versions 2, 3, and 4, that allows a local 3704 or 3705 controller to operate as an IBM 2701, 2702, or 2703 control unit (or any combination of the three) for certain data links, while performing network control functions for other links in the teleprocessing network.

**path.** (1) In VTAM, the intervening nodes and data links connecting a terminal and an application program in the host computer. (2) In SNA, the series of nodes, data links, and common network components (path control and data link control) that form the complete route traversed by the information exchanged between two network addressable units in session.

**PEP.** Partitioned emulation programming.

**physical unit.** (1) The control unit or cluster controller of an SNA terminal. (2) The part of the control unit or cluster controller that fulfills the role of a physical unit as defined by systems network architecture.

**positive response.** A response that indicates a message was received successfully. Contrast with *negative response*.

**program operator.** A VTAM application program that is authorized to issue network operator commands.

**protocol.** (1) In SNA, the sequencing rules for requests and responses by which network addressable units in a communication network coordinate and control data transfer operations and other operations. See also *bracket protocol*. (2) Synonymous with *line discipline*.

## Q

**queued for connection.** In VTAM, the state of a terminal that has logged on to an application program but has not yet been accepted by that application program. See also *connection*.

**quick closedown.** In VTAM, a closedown in which current data-transfer operations are completed, while new connection and data-transfer requests are canceled. Contrast with *orderly closedown*.

**quiesce protocol.** In VTAM, a method of communicating in one direction at a time. Either the VTAM application program or the logical unit assumes the exclusive right to send normal-flow messages, and the other node refrains from sending such messages. When the sender wants to receive, it releases the other node from its quiesced state.

## R

**record mode.** In VTAM, a mode of data transfer in which the application program can communicate with logical units or with local or remote 3270 Information Display Systems. Contrast with *basic mode*.

**remote.** In data communication, pertaining to devices that are connected to a data processing system through a data link.

**request.** (1) A directive that causes a data transfer or related operation to be performed. (2) In SNA, synonym for *request unit*.

**request header.** In SNA, a request/response header that indicates a request.

**request parameter list (RPL).** In VTAM, a control block that contains the parameters necessary for processing a request for data transfer, for connecting or disconnecting a terminal, or for some other operation.

**request/response header (RH).** In SNA, a control field, attached to a request/response unit (RU), that specifies the type of RU being transmitted—request or response—and contains control information associated with that RU. See also *request/response unit*.

**request/response unit (RU).** In SNA, the basic unit of information entering and exiting the transmission subsystem. It may contain data, acknowledgment of data, commands that control the flow of data through the network, or responses to commands.

**request unit.** In SNA, the request/response unit following a request header. Synonymous with *request*.

**responded output.** In VTAM, a type of output request that is completed when a logical unit receives a message and returns a response (if one is called for). Contrast with *scheduled output*.

**response.** (1) An answer to an inquiry. (2) The unit of information that is exchanged between VTAM or a VTAM application program and an SNA terminal to describe how a message arrived. (3) In SNA, synonym for *response unit*. (4) Contrast with *command, reply*.

**response header.** In SNA, a request/response header that indicates a response.

**response unit.** In SNA, the request/response unit following a response header; it is sent in response to a request unit. Synonymous with *response*.

**RH.** Request/response header.

**RPL.** Request parameter list.

**RPL-based macro instruction.** In VTAM, a macro instruction whose parameters are specified by the user in a request parameter list.

**RPL exit routine.** In VTAM, a user-written routine whose address has been placed in the EXIT field of a request parameter list. VTAM invokes the routine to indicate that an asynchronous request has been completed. See also *EXLST exit routine*.

**RU.** Request/response unit.

## S

**scheduled output.** In VTAM, a type of output request that is completed, as far as the application program is concerned, when the program's output data area is free. Contrast with *responded output*.

**SDLC.** Synchronous data link control.

**SDLC cluster controller.** A cluster control unit for a teleprocessing subsystem.

**sequence number.** A numerical value assigned by VTAM to each message exchanged between two nodes. The value (one for messages sent from the application program to the logical unit, another for messages sent from the logical unit to the application program) increases by one for each successive message transmitted unless reset by the application program.

**session.** (1) The period of time during which a user of a terminal can communicate with an interactive system; usually, the elapsed time from when a terminal user logs on the system until he logs off the system. (2) The period of time during which programs or devices can communicate with each other. (3) In SNA, a logical connection, established between two network addressable units (NAUs), that allows them to communicate. The session is uniquely identified by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session. See *LU-LU session, SSCP-LU session, SSCP-PU session*.

**session limit.** In the network control program, the maximum number of concurrent sessions on a non-SDLC, multipoint line.

**session parameters.** Synonym for *logon mode*.

**shared.** Pertaining to the availability of a resource to more than one user at the same time.

**simulated logon.** A logon generated for a terminal by VTAM at the application program's request. The application program accepts or rejects the terminal as if it had logged on. Synonymous with *application program logon*.

**single-thread application program.** A VTAM application program that processes requests from terminals one at a time. Such a program usually requests synchronous operations from VTAM, waiting until each operation is completed before proceeding. Contrast with *multithread application program*.

**SNA.** Systems network architecture.

**SNA terminal.** In VTAM: (1) A physical unit or logical unit. (2) A terminal that is compatible with systems network architecture.

**solicit.** In VTAM, to obtain data from a BSC or start-stop terminal or from a local 3270 terminal and move the data into VTAM buffers.



**solicited message.** A response from VTAM to a network operator command entered by a program operator. Contrast with *unsolicited message*.

**specific-mode.** In VTAM: (1) The form of read, receive, or solicit request that obtains data from one specific terminal. (2) The form of connection request that connects a specific terminal that has logged on. (3) Contrast with *any-mode*. See also *continue-specific mode*.

**SSCP.** System services control point.

**start options.** In VTAM, the user-specified or IBM-supplied options that determine certain conditions that are to exist during the time a VTAM system is operating. These options include: the size of VTAM buffer pools, which major and minor nodes are to be traced by the VTAM trace facility, and which major nodes are to be initially active. Start options can be predefined or specified by the network operator when VTAM is started.

**switched SNA major node.** In VTAM, a major node whose minor nodes are physical and logical units attached by switched SDLC links.

**synchronous operation.** In VTAM, a connection, communication, or other operation in which VTAM, after receiving the request for the operation, does not return control to the program until the operation is completed. Contrast with *asynchronous operation*.

**synchronous request.** In VTAM, a request for a synchronous operation. Contrast with *asynchronous request*.

**system services control point (SSCP).** In SNA, a network addressable unit that provides services via a set of command processors (network services) supporting physical units and logical units. The SSCP must be in session with each logical unit and each physical unit for which it provides services. It also provides services for the network operators or administrators who control the configuration. The SSCP is commonly located at a host node.

**systems network architecture (SNA).** The total description of the logical structure, formats, protocols, and operational sequences for transmitting information units through the communication system. Communication system functions are separated into three discrete areas: the application layer, the function management layer, and the transmission subsystem layer. The structure of SNA allows the ultimate origins and destinations of information—that is, the end users—to be independent of, and unaffected by, the specific communication-system services and facilities used for information exchange.

## T

**telecommunication network.** In a telecommunication system, the combination of all terminals and other telecommunication devices and the data links that connect them.

**telecommunication system.** In a teleprocessing system, those devices and functions concerned with the transmission of data between the data processing system and the remote users.

**teleprocessing subsystem.** In VTAM, a secondary or subordinate network and set of programs that are part of a larger teleprocessing system; for example, the combination consisting of an SDLC cluster controller, its stored programs, and its attached terminals.

**teleprocessing system.** A data processing system in combination with data communication facilities.

**terminal.** (1) A device, usually equipped with a keyboard and some kind of display, capable of sending and receiving information over a communication channel. (2) In VTAM, an end point in a telecommunication network; that is, a physical or logical unit, a start-stop or BSC device, or a 3270 Information Display System.

**terminal component.** A separately addressable part of a terminal that performs an input or output function, such as the display component of a keyboard-display device.

**transmission.** In data communication, one or more blocks or messages. For BSC and start-stop devices, a transmission is terminated by an EOT character. See also *block* and *message*.

**transmission control unit (TCU).** A communication control unit whose operations are controlled solely by programmed instructions from the computing system to which the unit is attached; no program is stored or executed in the unit. Contrast with *communications controller*.

## U

**unsolicited message.** A network operator message, from VTAM to a program operator, that is unrelated to any command entered by the program operator. Contrast with *solicited message*.

## V

**Virtual Telecommunications Access Method (VTAM).** A set of programs that control communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2.

**VTAM.** Virtual Telecommunications Access Method.

**VTAM application program.** Any program that uses VTAM macro instructions.

**VTAM definition.** The process of defining the communication network to VTAM and modifying IBM-defined VTAM characteristics to suit the needs of the user.

**VTAM definition library.** The DOS/VS files or OS/VS data sets that contain the VTAM definition statements and VTAM start options filed during VTAM definition.

## BIBLIOGRAPHY

This bibliography lists some of the non-VTAM IBM publications that are referred to in this publication or that relate to VTAM. (VTAM publications are shown in Figure P-1.) The listed publications (when available) contain a more detailed description of facilities that are used with VTAM.

### General Data Communications

*Introduction to Data Communications Systems*, SR20-4461

### Systems Network Architecture (SNA)

*Systems Network Architecture General Information*, GA27-3102

### Operating Systems

#### DOS/VS

*Introduction to DOS/VS*, GC33-5370

*DOS/VS System Generation*, GC33-5377 (includes storage estimates)

*DOS/VS System Management Guide*, GC33-5371

*DOS/VS System Control Statements*, GC33-5376

*DOS/VS System Utilities*, GC33-5381

*DOS/VS Serviceability Aids and Debugging Procedures*, GC33-5380

*DOS/VS and OS/VS TOLTEP for VTAM*, GC28-0663

#### OS/VS

*OS/VS Dynamic Support System*, GC28-0640

*DOS/VS and OS/VS TOLTEP for VTAM*, GC28-0663

#### OS/VS1

*OS/VS1 Planning and Use Guide*, GC24-5090

*OS/VS1 Storage Estimates*, GC24-5094

*OS/VS1 Services Aids*, GC28-0665

#### OS/VS2 SVS

*OS/VS2 SVS VTAM Component Release Guide*, GC27-0053

*OS/VS2 SVS VTAM Component Release Logic*, SY27-7279

#### OS/VS2 MVS

*OS/VS2 System Programming Library:*

*Service Aids*, GC28-0674

*Supervisor*, GC28-0628

*Overview*, GC28-0984

*Initialization and Tuning Guide*, GC28-0681

### IBM 3704 and 3705 Communications Controllers

*IBM 3704 and 3705 Control Program Generation and Utilities Guide and Reference Manual*, GC30-3008

*IBM 3704 and 3705 Communications Controllers: Network Control Program/VS;  
Program Logic Manual, SY30-3003*

*IBM 3704 and 3705 Communications Controllers: Network Control Storage and  
Performance Estimates (for OS and OS/VS TCAM and OS/VS and DOS/VS VTAM  
Users), GC30-3006*

*Introduction to the IBM 3704 and 3705 Communications Controllers, GA27-3051*

## INDEX

- abnormal termination (ABEND) 98
- ACB
  - control block
    - description 61
    - relationship to executable macro instruction 62
  - macro instruction 61
- acceptance
  - any mode 72
  - general description 71
  - specific mode 72
- accepting connection
  - description 13,71
  - to a specific terminal 72
  - to any terminal 72
  - to terminals 72
  - with a LOGON exit routine 72
- accounting exit routine
  - coding and installing 38
  - defining 38
  - executed inline 38
  - input to 38
  - run in supervisor state 38
  - task 38
- acquiring connection
  - CONALL option 72
  - CONANY option 72
  - general description 31,72
  - SIMLOGON 31
  - telecommunication security 124
- acquiring terminals that are connected to another program 72
- activating and deactivating
  - a remote communication controller 52
  - an NCP 51
    - description 47
    - effect on emulation mode 52
    - local SNA major nodes 51
    - local 3270s 50
    - non-existent terminals 55
    - PEP lines 52
    - remote attachments
      - example 53
      - general description 52
      - switched SNA major nodes 53
- activating duplicate names 19
- activating major and minor nodes as START options 39
- activating minor nodes during start processing 40
- activation, special considerations 54
- active
  - application program 54
  - terminal 54
- allocation of resources 1
- any mode 72,90
- APPL statement 21
- application program
  - active 54
  - as a network operator 56
  - as an ACB 5
  - authorization
    - acquiring terminals 120
    - block processing 120
    - effect on sharing resources 120
    - passing connection 120
    - program operator 120
  - authorized facilities 21
  - defined to VTAM 21,5
  - definition of 5
  - displaying status of 46
  - node structure 19
  - opening 71
  - starting and stopping 50
  - view of the network 9
  - writing 59
- application program-initiated connection 31
- application program-initiated disconnection 35
- APPLID field of ACB 19
- asynchronous request handling 65
- AT&T 83B3 Selective Calling Station 161
- ATCCONxx 39
- ATCCON00 39
- ATCSTRxx 39
- ATCSTR00 39
- audit trail 97,99
- authorization, application program 21
- authorization exit routine
  - coding and installing 37
  - defining 37
  - executed inline with VTAM 37
  - run in supervisor state 37
  - task for 37
  - telecommunication security 124
  - use in managing resources 120
- authorized facility, specified on APPL statements 21
- authorized path 109,70
- automatic dialing
  - for BSC and start-stop terminals 147
  - for SNA terminals 113
- automatic logon
  - altered by the network operator 55
  - description 31
  - effect of disconnection on 36
- auxiliary storage used by VTAM 8
- basic mode
  - communication 148
  - macro instructions
    - CHANGE 149
    - description 61
    - DO 148
    - READ 148
    - RESET 148
    - SOLICIT 148
    - WRITE 148
  - special I/O operations 150
  - special processing options 151
- Begin Bracket (BB) indicator 176
- Bid command 174
- Binary Synchronous Communication (BSC) terminals supported 162
- block processing
  - authorizing use of 21
  - effect on resource sharing 150
- bracket protocol
  - Begin Bracket indicator 87
  - Bid command 87
  - description 88
  - enforcement 88
  - Ready to Receive command 88
- BSC 3270 terminals communicating within record mode 89
- BTAM
  - compared to VTAM 135
  - using with non-SNA terminals 135
- buffering 40,102
- call-in terminals 146
- call-in/call-out terminals 147
- call-out terminals 147
- cancel closedown 45
- Cancel command 80,174

- chaining 80
- Change Direction Command indicator 85
- change direction indicators, summary 175
- change direction protocol
  - description 85
  - enforcement 86
  - indicators, summary 175
- Change Direction Request (REQ) indicator 175
- Change Direction Request indicator 86
- CHANGE macro instruction, description 149
- changing automatic logon 31
- channel adapter support 106
- character-coded (USS) definition table 26
- character-coded logoff 34
- character-coded logon
  - defining 26
  - description 26
  - IBM-defined 26
- Chase command 174
- CHECK macro instruction, description 62
- CID (see *communication identifier*)
- Clear command 176
- CLOSE macro instruction, description 60,64
- closedown procedure 36
- closing an application program 63
- closing VTAM down 36
- CLSDST macro instruction
  - description 60
  - PASS option 74
  - RELEASE option 74
  - use in disconnection 74
- cluster minor node, defining 22
- cold restart 51
- commands and indicators, summary 173
- commands expedited-flow 174
- commands normal-flow 174
- Communicating Magnetic Card SELECTRIC® Typewriter 160
- communicating with non-SNA terminals 148
- communication, basic mode 148
- communication common for record and basic modes 90
- communication controller
  - as a remote station 169
  - backup SDLC link 106
  - features not supported 106
  - features supported 106
  - general description 5
  - local and remote 105
  - models not supported 106
  - models supported 106
  - requirements for 105
  - 3705 two-channel support 106
- communication identifier (CID) 63,90
- communication macro instructions
  - basic mode 148
  - READ 148
  - RECEIVE 60
  - record mode 60
  - RESET 148
  - RESETSR 60
  - SEND 60
  - SESSIONC 60
  - SOLICIT 148
  - WRITE 148
- communication protocols
  - bracket protocol 87
  - change direction protocol 85
  - quiesce protocol 84
- communication, record mode 74
- communication, with the 3270 in record mode 89
- compatibility of VTAM and TCAM 128
- compatibility of local, BSC, and SNA 3270 terminals 177
- CONALL option 72
- CONANY option 72
- conditional disconnection 34
- confidential data, protecting 128
- configuration restart
  - delayed 104
  - immediate
    - of an NCP 103
    - of physical units 104
- connecting terminals 63
- connection
  - and terminal sharing 121
  - controlling 124
  - general description 71
  - macro instructions
    - CLOSE 60
    - CLSDST 60
    - OPEN 60
    - OPNDST 60
- continue-any mode 92
- continue-specific mode 92
- control block macro instructions
  - declarative 61
  - manipulative 61
- control blocks
  - ACB 61
  - EXLST 61
  - handling 64
  - NIB 61
  - relating to executable macro instructions 62
  - RPL 61
- controlling
  - a VTAM system 43
  - access to confidential data 128
  - access to VTAM 127
  - buffering 40
  - connections 125
  - non-SNA terminals 147
  - the VTAM system 12
  - use of VTAM facilities 127
- coordinating I/O 86
- counter overflow 97
- CPT-TWX Terminal 162
- CPU support 105
- creating a VTAM system 15,9
- data flow control 76
- data flow through a VTAM system 1
- data sets
  - for VTAM under DOS/VS 115
  - for VTAM under OS/VS 115
- DATAMGT macro instruction 15
- deactivation
  - forced 49
  - immediate 49
  - normal 49
  - restart 49
- declarative macro instructions 61
- defining
  - a character-coded (USS) definition table 27
  - an accounting exit routine 38
  - an authorization exit routine 37
  - application program major nodes 19
  - application program-initiated connection 31
  - automatic logons 31
  - BSC and start-stop terminals 22
  - BSC clusters 22
  - character-coded connection 26
  - connection procedures 25
  - control blocks, different methods 64
  - disconnection procedures 34
  - field-formatted connection 26
  - groups of lines 22
  - interpret tables 143
  - lines 22
  - local devices to the operating system 15

defining (continued)

- local SNA major nodes 22
- local 3270 major nodes 21
- logical unit pools 22
- logical units 22
- logon modes 32.1
- logon-interpret exit routines 144
- logons and logoffs 25
- major nodes 19
- minor nodes 19
- NCP major nodes 22
- network operator-initiated connection 32
- physical units 22
- ports 22
- SNA terminals 22
- start options filing 40
- start options predefined 40
- switched BSC and start-stop terminals 145
- switched line group 111
- switched SNA major nodes 23,111
- terminal components 22
- terminal-initiated connection 25
- terminal-initiated logons for non-SNA terminals 144
- terminals 22
- the network to VTAM 18
- VTAM buffering 40
- VTAM start options 39
- VTAM to the operating system 15
- VTAM-initiated connection 31

definite response

- requesting 78
- type 1 78
- type 2 78

definition macro instruction 23

definition statement 23

delayed configuration restart 104

delayed control mode 89

delayed request mode 89

designing VTAM application programs 12

DFASY exit routine 83

dial-in (see also *call-in*) 112

dial-out (see also *call-out*) 112

DISCNT parameter 35

disconnecting terminals 64

disconnection

- conditional 34
- effect on automatic logon 36
- general description 34,74
- of logical units 64
- of switched SNA terminals 114
- pass 74
- release 74
- requested by an application program 35
- requested by the network operator 36
- requested by the terminal 34
- requested by VTAM 36
- unconditional 34

DISPLAY command 46

displaying

- application programs 46
- NCPs 47
- physical units 47
- telecommunication lines 47
- terminals 46

distributed function 2

DO macro instruction, description 148

DOS/VS

- CPU support 105
- requirements 107

DSS (see *Dynamic Support System*)

dump utility for the NCP 99

dumps

- effect on emulation mode 114
- NCP 99
- operating system (OS/VS) 98

duplicate names 19

dynamic support system (DSS) 109

ECB compared to RPL exit routine 66

elements of a VTAM system 5

emulation mode

- effect of activation and deactivation 52
- not supported by VTAM 5

End Bracket (EB) indicator 176

end-of-day records 97

enforcement

- of bracket protocol 88
- of change direction protocol 86
- of quiesce protocol 85
- of session parameters 32

error

- circumventing 95
- detection and feedback 101
- notification 68
- recording 96
- recovery 102,95
- return codes 68

establishing procedures for using the system 12

establishing sets of session parameters 72

event control block (see *ECB*)

exception message 81

exception response, requesting 78

excess data, handling 93

EXECRPL macro instruction, description 62

EXLST control block, description 61

EXLST exit routines 67

- ATTN 67
- DFASY 67
- LERAD 67
- LOGON 67
- LOSTERM 67
- reenterability 67
- RELREQ 67
- RESP 67
- SCIP 67
- SYNAD 67
- TPEND 67

EXLST macro instruction, description 61

expedited message flow 76

expedited-flow commands 174

field-formatted logoff 34

field-formatted logon 26

filing NCP definition 23

filing VTAM definition statements 18

fixed storage pools 40

formatted system services (FSS) 26

four views of a VTAM system

- as viewed by an application program 9
- as viewed by the operating system 8
- as viewed by VTAM 9
- physical configuration 8

GENCB macro instruction, description 61

generating an NCP 18

HALT command 44

halting VTAM

- cancel 45
- designing TPEND for 45
- general description 44

- halting VTAM (continued)
  - orderly 45
  - quick 45
- handling excess data 93
- hardware error recording 97
- hardware error recovery procedures 102
- holding a physical unit connection 35
  
- I/O areas handling 64
- IBM 3270 communication in record mode 89
- identification verification
  - for BSC and TWX terminals 122
  - for physical units 122
  - for switched SNA terminals 122
  - for the host computer 122
- identifying terminals 90
- immediate configuration restart of an NCP 103
- immediate configuration restart of physical units 104
- immediate control mode 88
- immediate deactivation 36
- immediate request mode 89
- initial status of nodes 120
- Initiate Self command 25
- initiating requests for connection 55
- input receiving 81
- INQUIRE macro instruction, description 62
- installing a VTAM system 9
- interpret tables specifying 143
- INTRPRET macro instruction, description 62
  
- job names, relation to names of application programs 58
  
- KEEP option 93
  
- LDO control block 149
- LERAD exit routines 69
- levels of network control 43
- line scheduling specifications changing 56
- line sharing for start-stop and BSC terminals 121
- loading when activating an NCP 51
- local 3270, BSC, and start-stop terminal 8
- local 3270, defining 21
- local 3790, defining 22
- logical unit (see also *terminal*)
  - connection 25
  - defining to VTAM local 22
  - defining to VTAM non-switched 23
  - defining to VTAM switched 22
  - definition 7
  - minor node 22
  - pool 22
- Logical Unit Status (LUS) command 174
- logoff 34
- logon (see also *connection*) 25
- LOGON exit routine 25
- logon message 31
- logon mode name 33
- logon mode table 33
- logon-interpret routine 38,144
- logons controlling 124
- LOSTERM exit routine 67
  
- machine requirements 105
- macro instructions
  - basic mode 61
  - record mode 61
  - relating to control blocks 62
  - summary 59
  - ACB 61
- summary (continued)
  - CHECK 62
  - CLOSE 60
  - CLSDST 60
  - DO 148
  - EXECRPL 62
  - EXLST 61
  - GENCB 61
  - INQUIRE 62
  - INTRPRET 62
  - MODCB 61
  - NIB 61
  - OPEN 59
  - OPNDST 60
  - RCVCMO 60
  - READ 148
  - RECEIVE 60
  - RESET 148
  - RESETSR 60
  - RPL 61
  - SEND 60
  - SENDCMD 60
  - SESSIONC 60
  - SETLOGON 62
  - SHOWCB 61
  - SIMLOGON 60
  - SOLICIT 148
  - TESTCB 61
  - WRITE 148
  - types of 61
  - magnetic card typewriter, support of 161
  - major and minor node structure 19
  - major node 18
  - managing resources
    - through application programs 121
    - through NCP generation 120
    - through VTAM definition 119
  - manual switched CPU and communication controller backup 104
  - message
    - exception 81
    - flow, expedited 76
    - flow, normal 76
    - negative 76
    - positive 76
  - minor node 19
  - MODCB macro instruction, description 61
  - modifying an NCP definition 23
  - monitoring
    - application programs 46
    - NCPs 47
    - physical units 47
    - telecommunication lines 47
    - terminals 46
    - VTAM status 46
  - multiple console support 109
  - names, duplicate 19
  - names, reserved 19
  - naming nodes 19
  - NCP
    - defining to VTAM 110
    - general description 5
    - initial test 101
    - requirements 109
    - slowdown 103
  - network control macro instructions 60
  - network control mode 5
  - network control program (see *NCP*)
  - network operator
    - an application program as 56
    - considerations for control 57
    - general description 43

- network solicitor
  - how it works 140
  - messages 142
  - modifying 141
  - naming 141
  - password 142
  - release request 142
  - replacing 142
  - starting and stopping 55
- NIB macro instruction, description 61
- node structure major and minor 19
- non-SNA terminal
  - definition of 8
  - supported 159
- normal deactivation 36
- normal-flow commands 174
- normal message flow 76
  
- OPEN macro instruction, description 59
- opening an application program 63,70
- operating system requirements 107
- operating system traces 97
- OPNDST macro instruction description 60
- orderly closedown 45
- OS/VIS logon, security considerations 127
- OS/VIS requirements 108
- other telecommunication access methods
  - under DOS/VIS 132
  - under OS/VIS 133
- output responded 80
- output scheduled 80
  
- pacing 110
- pageable storage pools 40
- partitioned emulation programming extension (PEP)
  - effect of VTAM on 58.1
  - general description 5
  - planning considerations 114
- passing connection 31,124
- PEP (see *partitioned emulation programming extension*)
- permanent hardware errors 97
- physical unit 7
- procedures for the network operator 58.1
- program operator 56
- protecting confidential data 128
  
- queuing connection requests
  - effect on availability 71
  - general description 73
- quick closedown 45
- Quiesce at End of Chain (QEC) command 84,174
- Quiesce Complete (QC) command 84,174
- quiesce protocol 84
  
- RCVCMD macro instruction, description 60
- READ macro instruction, description 148
- Ready to Receive (RTR) command 88,174
- RECEIVE macro instruction, description 60
- receiving input 81
- record mode communication 74
- recording hardware errors 97
- recording software errors 97
- reenterability, requirements for EXLST exit routines 70
- Release Quiesce (RELQ) command 84,174
- reliability and availability support, summary 100
- reliability, availability, serviceability 95
- RELREQ exit routine 72
- remote devices not defined to the operating system 15
- remote terminals 106
- request mode selection 88
- Request Recovery (RQR) command 176
- Request Shutdown (RSHUTD) command 175
- request shutdown protocol 35
- requesting a response 78
- requesting another application program's terminal 72
- RESET macro instruction, description 148
- RESETSR macro instruction, description 60
- responded output 80,64
- response
  - negative 76,78
  - positive 76,78
  - requesting
    - definite response 78
    - exception response 78
- restart cold 51
- restart warm 51
- RPL macro instruction, description 61
  
- scheduled output 80,64
- security 121
- SEND macro instruction, description 60
- SENDCMD macro instruction, description 60
- sending messages
  - responded 80
  - scheduled 80
- sequencing 80
- serviceability aids
  - error recording 96
  - summary 96
- session control 76
- session parameters
  - general description 33
  - establishing 72
    - for local and BSC 3270 72
    - for non-SNA terminals 72
    - for SNA terminals 72
- SESSIONC commands 176
- SESSIONC macro instruction, description 60,76
- Set and Test Sequence Number (STSN) command 176
- SETLOGON macro instruction, description 62,72
- sharing resources
  - general description 2
  - managing 119
  - resources that can be shared 118
  - through logon 119
- SHOWCB macro instruction, description 61
- Shutdown (SHUTD) command 175
- Shutdown Complete (SHUTC) command 175
- Signal 175
- SIMLOGON macro instruction, description 61
- simulating a logon 32
- SNA terminals supported 155
- software error recording 97
- software errors, processing 103
- SOLICIT macro instruction, description 148
- solicitation
  - block 150
  - continuous 150
  - message 150
  - transmission 150
- specific mode 72,90
- Start Data Traffic (SDT) command 176
- start-stop terminals supported 159
- starting and stopping VTAM facilities 55
- starting VTAM
  - in DOS/VIS 44
  - in OS/VIS 44
- storage management 101
- storage pools
  - fixed 40
  - pageable 40
- storage requirements 107



storage sizes network operator considerations 58.1  
 supervisor state, accounting exit routine in 38  
 supervisor state, authorization exit routine in 37  
 support macro instructions 62  
 suppressing network operator messages 55  
 switched line group, generating 111  
 switched network support, network operator considerations 58.1  
 symbolic names used in controlling connection 127  
 SYNAD exit routines 69  
 synchronous request handling 65  
 system requirements 3  
 system services control point (SSCP) 26  
 System/3 166  
 System/32 158,167  
 System/370, as a supported terminal 167  
 System/7 (BSC) 167  
 System/7 (start-stop) 161

task for accounting exit routine 38  
 task for authorization exit routine 37

TCAM  
 altered operator control functions 131  
 functions not available in a shared environment 130  
 operator control functions replaced by VTAM functions 131  
 operator messages replaced by VTAM messages 132  
 use with VTAM 128

telecommunication security 121  
 teleprocessing online test executive program (TOLTEP) 99

terminal  
 active 54  
 connecting 64  
 disconnecting 64  
 non-SNA 8  
 sharing 121  
 SNA 7  
 supported 155

terminal-initiated connection for SNA terminals 25  
 terminal-initiated connection types 25

TESTCB macro instruction, description 61  
 TOLTEP (see *teleprocessing online test executive program*) 99

traces 97

transfer of data 1  
 TRUNC option 93

unconditional disconnection 34  
 unformatted system services (USS) 26  
 upward compatibility 107  
 using an interpret table with SNA devices 28

VTAM  
 compared to BTAM 135  
 CPU support 105  
 in operation 12  
 language 59  
 machine requirements 105  
 macro instructions 59  
 operating system requirements 107  
 operator commands  
 entering from a program operator 43  
 entering from the system console 43  
 summary 43  
 planning considerations 105  
 storage pools 40  
 system controlling 43  
 telecommunication security 121

traces  
 buffer traces 98  
 I/O traces 98  
 NCP line traces 98  
 storage management traces 98  
 use with TCAM 128  
 using with non-SNA terminals 135

warm restart 51  
 work areas handling 64  
 World Trade Telegraph 161  
 WRITE macro instruction, description 148  
 WU Plan 115A 162

1050 Data Communication System 159  
 2740 Communication Terminal Model 1 160  
 2740 Communication Terminal Model 2 160  
 2741 Communication Terminal 161  
 2770 Data Communication System 162  
 2780 Data Transmission Terminal 163  
 2980 General Banking Terminal System 162  
 3270 Information Display System (BSC) 164  
 3270 Information Display System (local) 155,159,105  
 3270 Information Display System (SNA) 156  
 3600 Finance Communication System 157  
 3650 Retail Store System 157  
 3660 Supermarket System 157  
 3735 Programmable Buffered Terminal 165  
 3740 Data Entry System 165  
 3750 Switching System 165  
 3767 Communication Terminal (SNA) 158  
 3767 Communication Terminal (start-stop) 161  
 3770 Data Communication System (BSC) 166  
 3770 Data Communication System (SNA) 158  
 3780 Data Communications Terminal 166  
 3790 Communication System (local) 156,105  
 3790 Communication System (remote) 158  
 5100 Portable Computer 161  
 5275 Direct Numerical Control Station 166  
 5937 Industrial Terminal (BSC) 166  
 5937 Industrial Terminal (remote) 158

VTAM Concepts and Planning

Virtual Telecommunications  
Access Method (VTAM)

READER'S  
COMMENT  
FORM

GC27-6998-3

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

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.

How did you use this publication?

- As an introduction
- As a reference manual
- For another purpose (explain) \_\_\_\_\_
- As a text (student)
- As a text (instructor)

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

Page Number:

Comment:

Cut or Fold Along Line

What is your occupation? \_\_\_\_\_

Newsletter number of latest Technical Newsletter (if any) concerning this publication: \_\_\_\_\_

If you wish a reply, give your name and address:

IBM branch office serving you \_\_\_\_\_

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.)

Reader's Comment Form

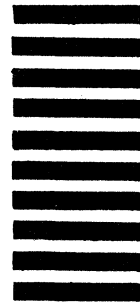
Cut or Fold Along Line

Fold

Fold

First Class  
Permit 40  
Armonk  
New York

**Business Reply Mail**  
No postage stamp necessary if mailed in the U.S.A.



Postage will be paid by:

International Business Machines Corporation  
Department 63T  
Neighborhood Road  
Kingston, New York 12401

Fold

Fold

VTAM Concepts and Planning (File No. S370-30) Printed in U.S.A. GC27-6998-3



International Business Machines Corporation  
Data Processing Division  
1133 Westchester Avenue, White Plains, New York 10604  
(U.S.A. only)

IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)

**This Newsletter No.** GN27-1545  
**Date** October 29, 1976

**Base Publication No.** GC27-6998-3  
**File No.** S370-30

**Previous Newsletters** None

### **VTAM Concepts and Planning**

© IBM Corp. 1974, 1975, 1976

This Technical Newsletter corresponds to VTAM Level 2 in DOS/VS, OS/VS1, and OS/VS2. It provides replacement pages for this publication. The replacement pages remain in effect for subsequent revisions of this publication unless specifically altered. The pages to be replaced are:

Title-Summary of Amendments	101-108
i-viii	127-128
31-32.1	161-164
39-60	185-192

A change to the text is indicated by a vertical line to the left of the change.

### **Summary of Amendments**

The attached Summary of Amendments contains descriptions of the technical and major editorial changes made by this Technical Newsletter.

**Note:** *Please file this cover letter at the back of the manual to provide a record of the changes that have been made.*



**This Newsletter No.** GN31-0606  
**Date** 31 May 1977  
**Base Publication No.** GC27-6998-3  
**File No.** S370-30  
**Previous Newsletters** GN27-1545

**VTAM Concepts and Planning**

© IBM Corp. 1974, 1975, 1976

This technical newsletter corresponds to VTAM Level 2 in DOS/VS, OS/VS1, and OS/VS2. It provides replacement pages for this publication. The replacement pages remain in effect for subsequent revisions of this publication unless specifically altered. The pages to be added or replaced are:

Title page, Edition notice	79,80
Summary of Amendments/Summary of Amendments	87,88
vii, viii	105-108
19,20	127-130
43-46	155-168
	187-192

A change to the text is indicated by a vertical line to the left of the change. A change to a figure is indicated by a vertical line to the left of the figure caption.

**Summary of Amendments**

The attached Summary of Amendments dated 31 May 1977 describes the technical and major editorial changes made by this technical newsletter.

**Note:** *Please file this cover letter at the back of the manual to provide a record of the changes.*





# Technical Newsletter

**This Newsletter No.** GN31-0890  
**Date** December 15, 1978  
**Base Publication No.** GC27-6998-3  
**File No.** S370-30  
**Previous Newsletters** GN27-1545  
GN31-0606

## VTAM Concepts and Planning

©IBM Corp. 1974, 1975, 1976

This Technical Newsletter corresponds to VTAM Level 2 in OS/VS2 MVS. It provides replacement pages for this publication. The replacement pages remain in effect for subsequent revisions of the publication unless specifically altered. The pages to be added, replaced, or removed are:

Title, Edition Notice  
Summary of Amendments  
i, ii  
107, 108  
113, 114  
185, 186

A change to the text is indicated by a vertical line to the left of the text.

### Summary of Amendments

The attached Summary of Amendments dated December 15, 1978, contains a description of the technical changes made by this technical newsletter.

**Note:** *Please file this cover letter at the back of the manual to provide a record of the changes that have been made.*





**IBM**

**International Business Machines Corporation  
Data Processing Division  
1133 Westchester Avenue, White Plains, New York 10604  
(U.S.A. only)**

**IBM World Trade Corporation  
821 United Nations Plaza, New York, New York 10017  
(International)**